THE UNIVERSITY
of EDINBURGH

Quantitative tool for *in vivo* analysis of DNA-binding proteins using High Resolution Sequencing Data

Milana S. Filatenkova

Doctor of Philosophy

The University of Edinburgh

2016

Declaration

I declare that this thesis has been composed solely by myself and that it has not been submitted, either in whole or in part, in any previous application for a degree. Except where otherwise acknowledged, the work presented is entirely my own.

Milana Filatenkova

March 2016

Abstract

DNA-binding proteins (DBPs) such as repair proteins, DNA polymerases, recombinases, transcription factors, etc. manifest diverse stochastic behaviours dependent on physiological conditions inside the cell.

Now that multiple independent *in vitro* studies have extensively characterised different aspects of the biochemistry of DBPs, computational and mathematical tools that would be able to integrate this information into a coherent framework are in huge demand, especially when attempting a transition to *in vivo* characterisation of these systems.

ChIP-Seq is the method commonly used to study DBPs *in vivo*. This method generates high resolution sequencing data – population scale readout of the activity of DBPs on the DNA. The mathematical tools available for the analysis of this type of data are at the moment very restrictive in their ability to extract mechanistic and quantitative details on the activity of DBPs. The main trouble that researchers experience when analysing such population scale sequencing data is effectively disentangling complexity in these data, since the observed output often combines diverse outcomes of multiple unsynchronised processes reflecting biomolecular variability.

Although being a static snapshot ChIP-Seq can be effectively utilised as a readout

for the dynamics of DBPs *in vivo*. This thesis features a new approach to ChIP-Seq analysis – namely accessing the concealed details of the dynamic behaviour of DBPs on DNA using probabilistic modelling, statistical inference and numerical optimisation. In order to achieve this I propose to integrate previously acquired assumptions about the behaviour of DBPs into a Markov-Chain model which would allow to take into account their intrinsic stochasticity. By incorporating this model into a statistical model of data acquisition, the experimentally observed output can be simulated and then compared to *in vivo* data to reverse engineer the stochastic activity of DBPs on the DNA.

Conventional tools normally employ simple empirical models where the parameters have no link with the mechanistic reality of the process under scrutiny. This thesis marks the transition from qualitative analysis to mechanistic modelling in an attempt to make the most of the high resolution sequencing data.

It is also worth noting that from a computer science point of view DBPs are of great interest since they are able to perform stochastic computation on DNA by responding in a probabilistic manner to the patterns encoded in the DNA. The theoretical framework proposed here allows to quantitatively characterise complex responses of these molecular machines to the sequence features.

Acknowledgements

Special thanks to Vincent Danos and Meriem El Karoui for supervising my PhD and supporting me along this difficult path. They introduced me to the exciting field of computational biology and helped me identify where my strengths lie.

Thanks to Charlie Cockram and David Leach for their contribution of ChIP-Seq data, which allowed me to test the data analysis tool proposed in this thesis.

Also, I would like to thank Nacho Molina, Natalia Bochkina and Philipp Thomas for valuable suggestions regarding the mathematical part of my PhD.

Finally, I thank the Darwin Trust (PhD scholarship) and the European Research Council (Advanced Grant RULE-320823) for financially supporting this project.

Contents

List of Tables

List of Figures

Abbreviations

DBP	DNA Binding Protein
DSB	Double Strand Break
MC	Markov Chain
AS	Absorbing State
SA	Stochastic Automaton
SM	Stochastic Machine
HR	Homologous Recombination
HRSD	High Resolution Sequence Data
ChIP-Seq	Chromosome Immuno-Precipitation Sequencing
PCR	Polymerase Chain Reaction
NB	Negative Binomial Distribution
MN	Multinomial Distribution
MLE	Maximum Likelihood Estimator
LR	Likelihood Ratio
LRT	Likelihood Ratio Test
RWM	Random-Walk Metropolis
BIC	Bayesian Information Criterion
OS	Segment Output
ID	Ideal Data
IPD	Ideal Pileup Data
RPD	Real Pileup Data

Chapter 1

Introduction

There is a wide range of enzymes known to perform their function by interacting with the DNA (DNA-binding proteins). They are repair proteins, DNA polymerases, replicases, nucleases, recombinases, transcription factors and histones. DNA-binding proteins (DBP) are involved in important cellular processes such as recombination, replication and transcription. Experimental evidence suggests that the interaction of some of these proteins with the DNA is mediated by the underlying sequence base composition (sequence-specific interaction) (reviewed by Rohs *et al.* (2010)). This study will specifically focus on such DBPs.

The interactions of these proteins with the DNA are commonly characterised *in vivo* by chromatin immunoprecipitation (ChIP). ChIP-Seq provides the best resolution among other ChIP methods combining chromatin immunoprecipitation (ChIP) with DNA sequencing to identify the binding sites of DBPs (for a review see Furey (2012)). In these experiments DNA bound to the protein of interest is isolated, fragmented, sequenced and then mapped to the reference genome to identify the location of the protein binding on the sequence. Afterwards the sequence footprints of multiple unsynchronised species of DBP coming from a

large population of cells are pulled together to generate an analysable output (ensemble average signal). ChIP-Seq generates a very noisy signal which should be interpreted as frequency of a nucleotide occurrence in the population of sequence measurements.

It appears that this type of sequencing data is extremely difficult to analyse (especially when the goal is to extract quantifiable characteristics of DBPs of interest) due to the complexity of the acquired data: the large scale (multiple proteins' footprints are pulled together), the significant noise present in the data and most importantly the non-deterministic nature of DBPs, i.e. their functional variability. DBP stochasticity manifests itself in their ability to randomly switch between conformational states, either spontaneously or as a result of an interaction with a substrate (and the reaction at this scale is highly stochastic) resulting in functional heterogeneity, which is not so straightforward to account for when analysing the data.

Since DBP activity on the DNA is intrinsically stochastic, the data would contain in itself traces of a whole range of different functional behaviours across a population of DBP species and hence a mixture of multiple fragmented sequence segment outputs of their diverse activity. In order to make the most of population average sequencing data we have to consider a variety of possible kinetic strategies of a DBP of interest under a stochastic model (because DBPs behave stochastically). This is particularly important when the goal is to extract mechanistic insight into DBP activity on the DNA rather than simply pointing out likely positions of DBPs on the sequence.

Among the current methods of ChIP-Seq analysis are MACS (Zhang *et al.*, 2008a), PeakSeq (Rozowsky *et al.*, 2009a), SAGE (NB) (Robinson & Smyth, 2007), RNA-Seq (NB) (Robinson *et al.*, 2010), BayesPeak (Spyrou *et al.*, 2009), MOSAiCS (Kuan *et al.*, 2009) and others. The reality is that these algorithms

are not sophisticated enough in the sense that although they allow to identify the locations where the protein can bind with a certain degree of confidence, they lack the ability to extract quantitative information about the stochastic behaviour of DBP on the DNA. Therefore there is need for a quantitative tool that would allow us to make advantage of these population average genomic data by unmasking quantifiable properties of DBPs manifested *in vivo*.

The aim of this thesis is to build a novel tool, tailored to extract quantitative mechanistic details about DBP activity *in vivo* from large scale population average sequencing data like ChIP-Seq. This tool will combine a probability based model in order to account for the variability in DBP behaviour (this model will be constructed based on the initial assumptions about DBP mechanism), with DNA sequence input and statistical inference to infer the parameters of the model. The idea is to reverse engineer the sequence of transition states of DBP and then incorporate the distribution over the reachable states (the probability distribution of the sequence outputs generated by the model) into a statistical model of the ChIP-Seq experiment, and then compare the model output with the real sequencing data acquired *in vivo*. The model comparison with the data will allow us to test the assumptions of the model and estimate its parameters.

Herein I focus particularly on an *E.coli* double strand break (DSB) repair molecule called RecBCD (see Dillingham & Kowalczykowski (2008) for a review) as an example of DBP and analyse its mechano-genomic properties using ChIP-Seq data generated by the sequence footprints of a population of RecBCDs. Though this method has been developed specifically for RecBCD it can be well extrapolated to *in vivo* analysis of other types of DBPs provided one has access to the data of their sequence footprints such as ChIP-Seq.

Certain aspects of RecBCD activity on the DNA have been revealed and some key parameters measured *in vitro* using single-molecule techniques such as the

resection rate, processivity, interaction with special sequence motifs (Chi sites), backsliding and pausing, and the influence of the experimental conditions on these parameters (as reviewed in Carrasco *et al.* (2014)). The conditions in the cell may be different though from those created in the test tube, resulting in differing behaviour of this enzyme, potentially exhibiting other unpredictable types of behaviour unseen in the conditions of a test tube. It would be of interest to test how the mechanism differs in the real physiological conditions of the cell.

Making an assumption that recombinase protein (RecA) fully covers the sequence output generated by RecBCD as a result of its resection of a DSB, ChIP-Seq data of RecA binding will be used as a readout of RecBCD activity *in vivo*.

I will construct a Markov Chain (MC) framework of RecBCD activity on the DNA by integrating existent knowledge and assumptions (obtained from the literature) into the state diagram of a Markov process. The "readout" protein (RecA) has been known to stay bound to the DNA for a very long time after RecBCD has completed its job and dissociated from the DNA, this time being much longer than the time in the transition phase. For this reason, I will use a special type of MC namely Absorbing MC. In this framework the global state of dissociation of RecBCD will be mapped to an absorbing state of the associated MC. We shall be interested only in the output produced in the absorbing state (AS) of the process, since being the most long-lived it will generate the largest contribution to the measured output. In fact the contribution of the transient states will be negligible compared to AS. Thus, the idea is to incorporate the output of the Markov Chain produced in the absorbing state as a prior model into a statistical model of ChIP-Seq data, which will also be developed, tested and discussed in this thesis.

The MC model proposed in this thesis will have a parametric form, meaning the structure of the model will depend on the few key parameters known to

govern RecBCD function such as processivity, probability of Chi recognition and macroscopic motor speeds. The likelihoods of the data under individual sets of values of these parameters will be compared in order to select the one for which the data is most likely (Maximum likelihood method). Should the most likely model fail to fit the data this would be indicative of the presence of additional factors that have been previously disregarded, which would need further exploration, potentially using *in vitro* single molecule techniques given the immense complex and time consuming *in vivo* experimental setup.

Here I also explore and extend a general formulation of stochastic computation on the DNA for those DBPs that "read" the DNA and respond to special sequence motifs (such as Chi sites) in a probabilistic manner, RecBCD being one of them (see Touzain *et al.* (2010) for a review). These DBPs can be viewed as "stochastic computers" operating on the DNA. In the first chapter of this thesis I will utilise the notion of Stochastic Automaton (SA) to describe the behaviour of DBPs.

A Stochastic Automaton is a mathematical model for a system that has a finite number of states. By taking in an input σ it performs a transition from current state s into one of the possible states s_i after a discrete time interval has lapsed, and the probability of this transition depends on both current state s and input σ - $p(s, \sigma)$. The transition is associated with an output that depends on s_i. Thus within its life cycle the SA translates a sequence of inputs into a sequence of outputs (Rabin, 1963).

Bennett (1982), Adar *et al.* (2004), Bar-Ziv *et al.* (2002), Benenson *et al.* (2001) and Benenson *et al.* (2003) introduced the notion of molecular SA, which can be any molecule (an enzyme for example) that takes an input and produces an output both in a molecular form to perform computation. The SA framework is preferred when dealing with molecular stochasticity.

As for DNA processing molecules, Benenson *et al.* (2003) and Adar *et al.* (2004) applied SA framework to DNA restriction and ligation enzymes that use the DNA as both "input" and"software" (encoding transition rules) to produce DNA "output" and using ATP as fuel. In their model each computational step of the automaton consists of reversible self-assembly of the "hardware molecule" on the input DNA. They consider only two states of the molecules - either bound to the DNA or free. The SA performs its computation by recognising the sequence GGATG and cleaves 9 and 13 nucleotides away from the recognition site. They refer to the recognition sites as "software" and say the cleavage is "software directed". The transition probabilities of the automaton are governed by the concentrations of the transition molecules and "software" molecules.

Another example of a DNA computer (also referred to as "sequence reading" machine) is the RecA assembly cascade on the DNA that carries out another type of computation - the discrimination of close-by sequences (Bar-Ziv *et al.*, 2002). The probabilities of state transitions (RecA binding events) are encoded by the specific sequence triplets.

In principle, the transition rules of an automaton should be constructed so as to incorporate both prior biochemical knowledge about the enzyme and the composition of sequence. The output of the computation is a probability distribution over the final states rather than a single final state.

This thesis is organised as follows:

- Chapter 2, called "Preliminaries" is the summary of the preliminary information used in the following chapters. It features biological foundations of the mechanism of DNA double strand break repair in *E.coli*. It also reviews the role of RecBCD in repair and the mechanistic model of its DNA processing activity based on knowledge obtained *in vitro*. It also gives an

introduction to ChIP-Seq: description of the experimental steps and data acquisition. And finally, Chapter 2 provides an overview of mathematical methods used in the thesis, such as probability models, Markov Chain, Maximum Likelihood, numerical methods of optimisation, etc.

- The MC model of a "sequence switchable" stochastic automaton and the distribution over its absorbing states are derived in Chapter 3. This model is inspired by RecBCD system but can be easily extrapolated to describe other DNA processing machineries. Also, it demonstrates some useful results concerning the mean output and how it depends on the constraints of the model.

- In Chapter 4, the distribution over the absorbing states of MC derived in Chapter 2 is incorporated into a model of ChIP-Seq data acquisition. In this Chapter I only consider ideal data devoid of the limitations pertaining to a real experiment. I also construct an objective function that enables robust estimation of the parameters of the MC on these ideal data. Finally, I prove that this function allows to recover the parameters used to simulate the synthetic "ideal" data, in the limit when the population size is infinitely large (Chis *et al.*, 2011).

- Chapter 5 provides an analysis of parameter sensitivity to the constraints imposed by the experimental setup, the parameters being estimated using the objective function derived in Chapter 4.

- Chapter 6 is a case study where the framework developed in the preceding chapters is applied to ChIP-Seq data of RecBCD-mediated RecA binding in the vicinity of DSB in order to test the mechanistic assumptions about RecBCD activity proposed *in vitro*.

- Chapter 7 provides an overview of the results obtained in Chapters 3-6. It

also discusses the limitations of the mathematical framework proposed in this thesis and suggests potential ideas for future work.

Chapter 2

Preliminaries

2.1 Biomolecular background

2.1.1 Double strand break in *E.coli*

Naturally, double strand breaks (DSB) in *E.coli* are formed at a rate of approximately two per cell per hour (Vilenchik & Knudson, 2003). One of the most common sources of DSB in *E.coli* is when a replication fork passes by a single strand gap, and by copying this gap leaves a double stranded end (see Fig. 2.1). This is also referred to as "replication fork collapse" (Dillingham & Kowalczykowski (2008) for a review). Double strand break repair (DSBR) is essential for the chromosome replication to be completed before duplication of the chromosome and for cell survival accordingly. Cells whose repair machinery is inactive are very sensitive to double strand damage. One unrepaired DSB per replication cycle is lethal for the cell (Eykelenboom *et al.*, 2008).

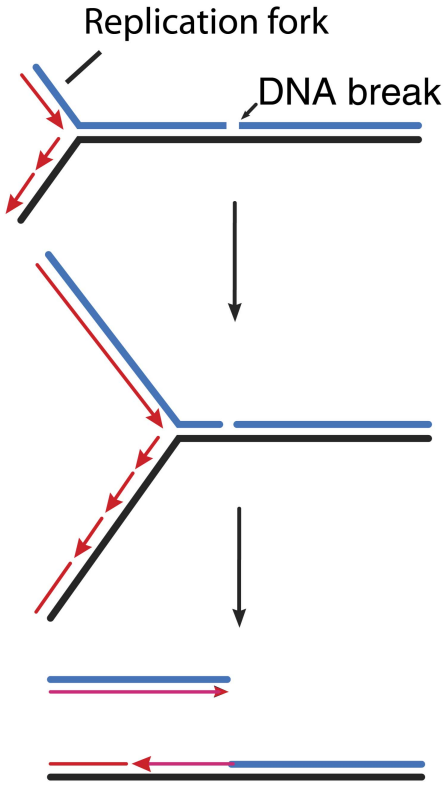


Figure 2.1. How double strand breaks occur during replication (derived from Cox (2013))

2.1.2 RecBCD and repair of double strand breaks by homologous recombination

Two strategies of DSB repair exist in *E.coli*: non-homologous end joining and homologous recombination. Here, we shall focus on the repair by homologous recombination (HR) initiated by RecBCD protein complex (Symington & Gautier, 2011).

DSB repair by HR requires a DNA donor to complete repair (for a review see Wyman *et al.* (2004)). In HR the missing information is copied from the donor-intact chromosome which serves as a template to fill in the missing nucleotides in the gap formed upon double strand break. The present state of knowledge based

mostly on *in vitro* studies is that RecBCD-mediated double strand break repair begins with RecBCD binding to the DSB end. Then the complex rapidly advances along the DNA away from the location of the break (Fig. 2.2). According to the uncoupled translocation model the two motors of RecBCD - RecB and RecD - move independently on the individual strands of the DNA, motor D being significantly faster than motor B. At the same time both strands of DNA are degraded behind the complex by a nuclease subunit (RecB). Resection of the double strand end by RecBCD proceeds by degradation of both strands in an asymmetric manner until a hotspot instigator ("Chi" site: 5'-GCTGGTGG-3') is recognised, thereby producing a single strand 3' overhang. This single stranded intermediate mediates invasion into a homologous strand and pairing with a complementary segment of DNA located on the intact homologous chromosome. The pairing is catalysed by the recombinase RecA which forms a protofilament around the 3' single stranded DNA. This single stranded intermediate displaces a single strand loop upon invasion into an intact chromosome. PriA catalyses synthesis of DNA using the displaced single strand as a template. The points of single strand crossover are called Holliday Junctions (HJ). The intertwined chromosomes are resolved by cleaving HJ, which leads to the creation of two intact DNA molecules.

DSB processing by RecBCD before the stage of homologous recombination performed by helicases and nucleases can be summarised in three essential steps (Fig. 2.3):

- DSB recognition by RecBCD.
- Long range end resection by RecBCD.
- Recombinase (RecA) loading on the 3' overhang.

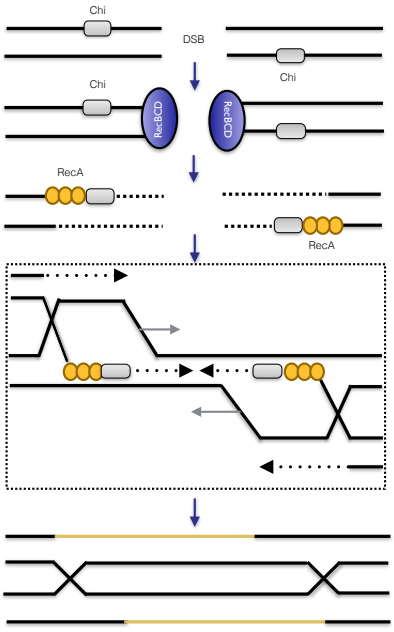


Figure 2.2. RecBCD mediated repair of double strand breaks by homologous recombination

Kinetics of DSB repair measured at 30°C (Lesterlin *et al.*, 2014):

- Overall, it takes 150 *min* to repair a DSB.
- RecA bundles start forming 5 *min* after DSB and reach maximum after 15 *min*.
- Homologous search lasts for 47 *min*.
- Pairing takes 5 *min*.
- The bundles are disassembled after 17 *min*.

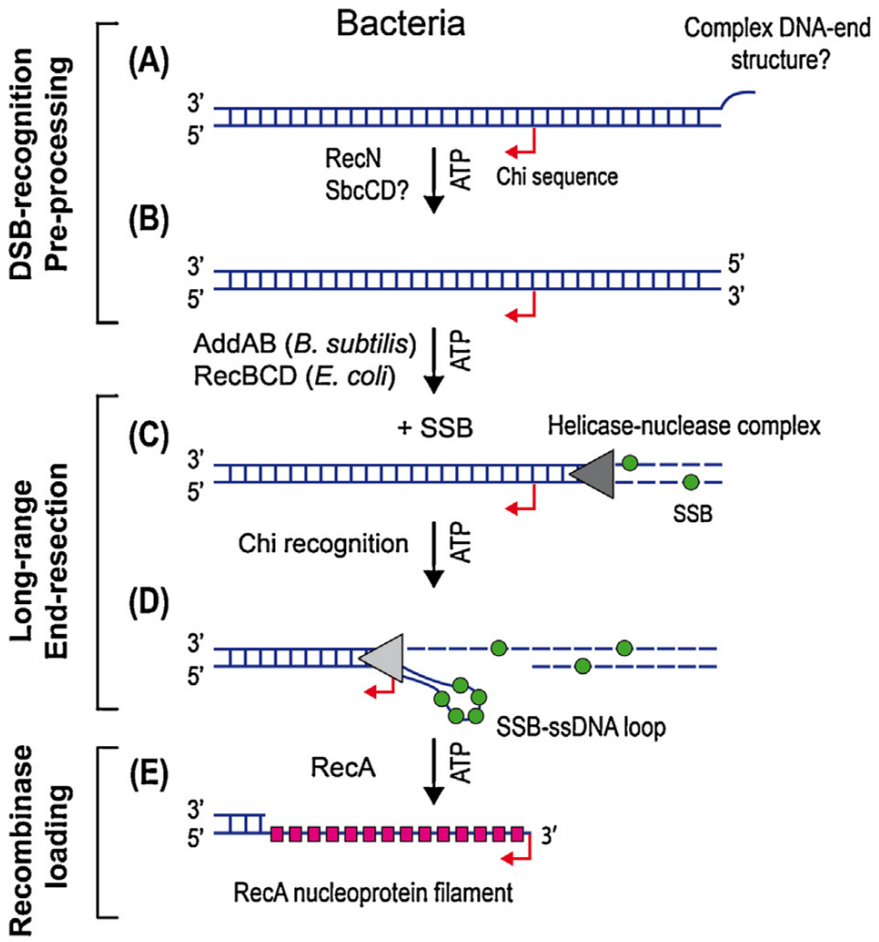


Figure 2.3. Stages of DSB processing prior to Homologous Recombination for prokaryotic and eukaryotic systems. (A) A DSB containing a potentially complex DNA-end structure. (B) The complex-DNA structure is trimmed to a blunt or nearly-blunt end by the action of nucleases. (C) Helicases and nucleases work in a coordinated manner to unwind and cleave the duplex DNA upstream of Chi. Single-stranded binding (SSB) proteins help to stably separate the two strands. (D) Downstream of Chi the 3'-end is no longer cleaved and protected by SSB proteins. (E) Recombinase RecA displaces SSB protein and forms a nucleoprotein filament suitable for strand exchange in homologous recombination (reprinted from Carrasco *et al.* (2014)).

2.1.3 RecBCD crystal structure

RecBCD is a supramolecular protein complex, containing three individual intertwining subunits: RecB, RecD & RecC (see Fig. 2.4). Its crystal structure was solved by Singleton *et al.* (2004). RecB consists of two parts - one helicase and a nuclease, at the same time being able to unwind duplex DNA using ATP and degrade both single strands in the presence of Mg^{2+} and Ca^{2+}; RecD is a leading helicase. RecC has a groove - Chi scanning site, that upon interacting with a Chi sequence triggers conformational change in the RecBCD complex.

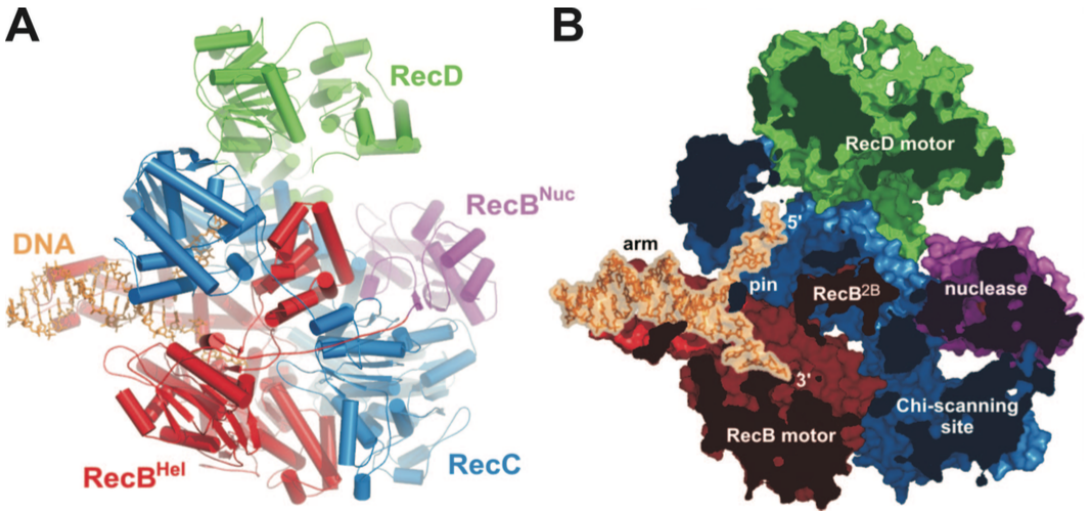


Figure 2.4. The RecBCD-DNA complex (Singleton *et al.*, 2004)

2.1.4 Early stage: zoom in on RecBCD dependent DSB resection

Double strand break resection by RecBCD is a multistep process, triggered by RecBCD loading on a DSB-end. The mechanistic model of RecBCD activity has been developed based on the multiple *in vitro* biochemical and single molecule

studies (Carrasco *et al.* (2014) for a review). The process of formation of a single stranded overhang that stimulates strand exchange consists of three major parts - RecBCD propagation along the DNA before it encounters a Chi site (Fig. 2.5 - A), the moment when RecBCD changes its mode upon interaction with a Chi site (Fig. 2.5 - B) and RecBCD (Fig. 2.5 - C) trajectory after recognition of a Chi site. RecBCD uses both its helicases with opposite polarity RecB $3' \rightarrow 5'$ and RecD $5' \rightarrow 3'$ to unwind duplex DNA. RecD is the leading helicase moving along the 5' strand and RecB is a lagging helicase advancing on the 3' strand with a speed twice less than that of RecD (Taylor & Smith, 2003). The asymmetric unwinding leads to the accumulation of a single stranded loop ahead of the complex on the 3' side (Spies *et al.*, 2003). As RecBCD propagates along the DNA RecB nuclease degrades both single strands behind the complex. RecB motor not only participates in unwinding of the duplex DNA but also plays an important role in feeding the 3' strand to RecC Chi recognising site. When a Chi sequence passes through the groove inside RecC subunit (Arnold *et al.*, 2000; Handa *et al.*, 1997) recognition occurs in a stochastic manner. Upon encountering a Chi site, RecC subunit binds tightly to the 3' tail, preventing further digestion of this strand (Singleton *et al.*, 2004) as the strand is no longer accessible to the nuclease domain. Chi recognition also leads to a pause at Chi with subsequent conformational change in the complex, leading to inactivation of RecD helicase. RecB taking over the leading helicase activity, the complex now moves with a reduced speed (Handa *et al.*, 2005; Spies *et al.*, 2003), two-fold slower on average (Spies *et al.*, 2007). Conformational modification occurs in RecB - nuclease subunit, whereby RecB is no longer capable of degrading the 3' strand, the final cleavage event being at Chi (Taylor & Smith, 1995). The RecB conformational change also entails subsequent exposure of its RecA nucleating surface stimulating nucleation of RecA (Spies *et al.*, 2007) which gets transferred to the ssDNA forming protofilament to catalyse strand invasion. Thus RecBCD combines helicase/nuclease and recombinase-loading activities. Modelling of the

interaction between RecA and the nuclease domain of RecB suggests that it is similar to the RecA-RecA interface in the nucleoprotein filament (Spies & Kowalczykowski, 2006). Now, RecBCD continues advancing on DNA using RecB helicase to unwind the DNA all the way until RecBCD falls off the DNA. The end product is the single stranded DNA covered by RecA filament.

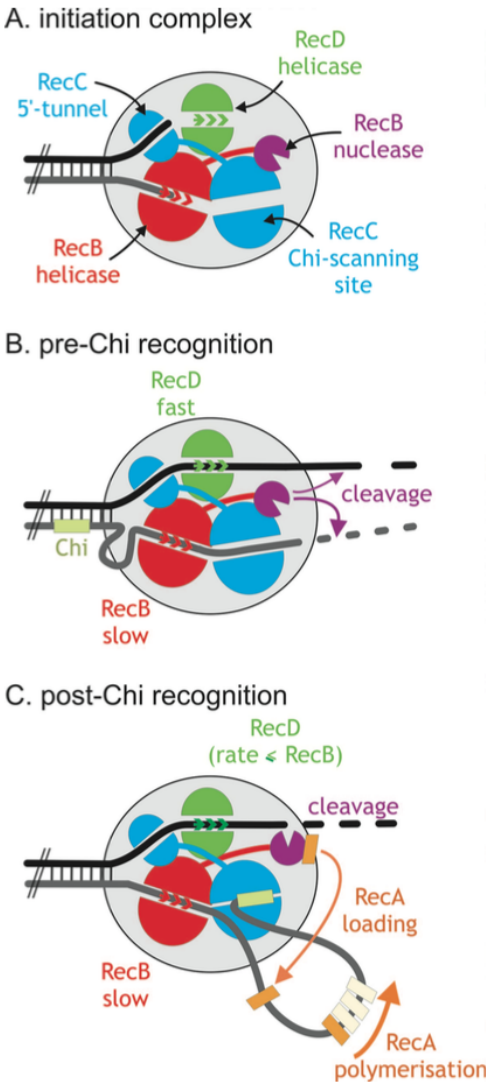


Figure 2.5. Model for RecBCD enzyme mechanism (Dillingham & Kowalczykowski, 2008)

2.1.5 Current understanding of RecBCD response to a Chi site

RecB is needed to pass a single strand through the Chi-recognising site of RecC (Spies *et al.*, 2005). Mutation of RecB impedes the movement of a single strand through RecC (Taylor & Smith, 2003). When a Chi-site passes through the groove inside the RecC subunit (Arnold *et al.*, 2000; Handa *et al.*, 1997) recognition occurs in a stochastic manner. Having encountered a Chi site, RecC subunit binds tightly to the 3' tail, preventing further digestion of this strand (Singleton *et al.*, 2004). Chi recognition also leads to a pause at Chi (Handa *et al.*, 2005; Spies *et al.*, 2003). The distribution of pause durations follows an exponential decay with lifetimes of 3.5 ± 0.3 s and 3.9 ± 0.2 s and was suggested to be a result of the conformational change of RecBCD complex (Spies *et al.*, 2007). Also following Chi recognition and conformational change a linker opens to let the single stranded loop through (Wigley, 2012; Yang *et al.*, 2012) (Fig. 2.6). The other consequence of Chi recognition is a change in the translocation rate and inactivation of RecD. The loop could potentially be important to prevent rejoining of the single strands of the unwound DNA, however it is not absolutely essential. For example, RecBC is able to perform the function of RecBCD without forming a loop (Taylor & Smith, 2003). The formation of the 3' ssDNA loop could also be related to the underlying longer-lived Chi-RecC subunit interaction that mediates the Chi-induced enzymatic changes in RecBCD (Dillingham & Kowalczykowski, 2008). The recognition of a Chi-site is a probabilistic process - the estimated probability of Chi recognition *in vitro* is only $\sim 20 - 40\%$ (Dixon & Kowalczykowski, 1993a). Dwell times of a Chi-site at the binding locus are extremely small (< 1 ms) with a single chance to achieve recognition (Carrasco *et al.*, 2014). It has been suggested that when Chi passes through the key amino acids of RecC responsible for Chi recognition a "battle" occurs between the

helicase activity (translocation) and Chi-recognition (pausing) (Carrasco *et al.*, 2014) (Fig. 2.7). RecC can also interact with Chi-like sequences - single-base variants of Chi sequence. Pausing at Chi-like sequences has been confirmed for example by Yang *et al.* (2012), but those interactions should have even a lower probability in order to result in a conformational change.

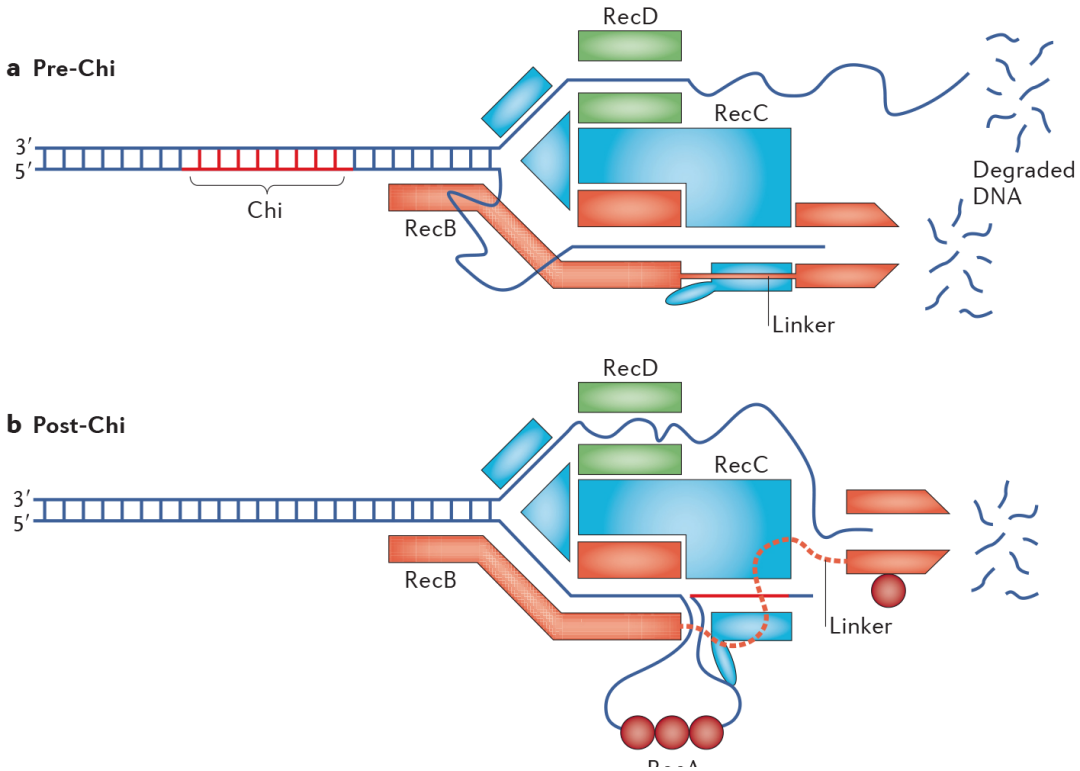


Figure 2.6. A model for the current understanding of how RecBCD responds to Chi (Wigley, 2012)

2.1.6 RecBCD processivity

RecBCD unwinds duplex DNA in discrete steps, with an average unwinding "step-size" $m = 3.9(\pm 1.3)\ bp/step$, with an average unwinding rate of $k_U = 196(\pm 77)\ step/s$ ($mk_U = 790(\pm 23)\ bp/s$) at 25°C (Lucius *et al.*, 2002). RecBCD

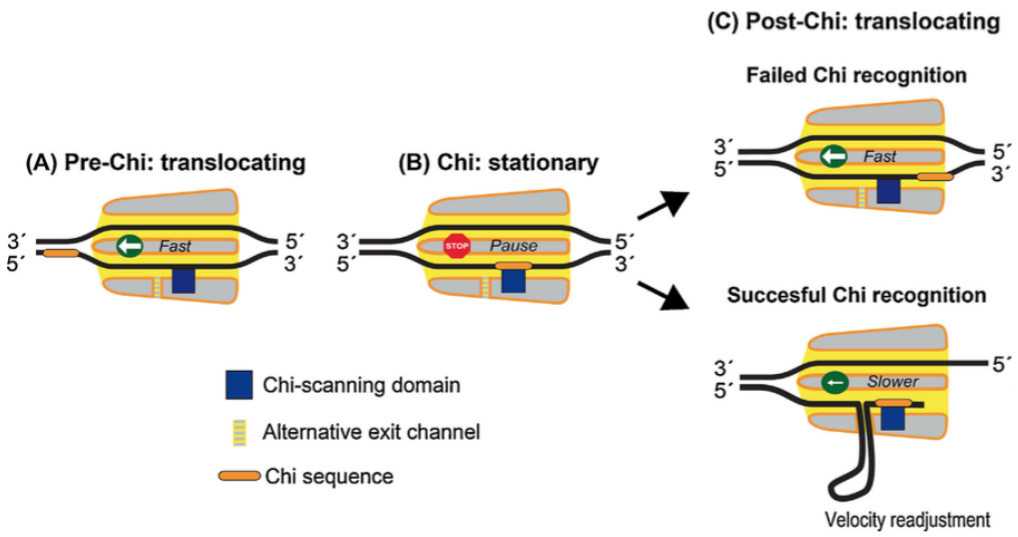


Figure 2.7. Generalized model for pausing and loop formation induced by Chi recognition in bacterial helicase-nucleases (Carrasco *et al.*, 2014).

is a very processive helicase, being able to unwind 30 *kbp* on average as measured *in vitro* (Roman *et al.*, 1992) before dissociating from the DNA. Its processivity has also been found to depend on the overall speed (Spies *et al.*, 2007).

The distance over which Chi acts in cells was estimated to be roughly an exponential with 50% drop at about $2-4$ *kb* (Cheng & Smith, 1989; Ennis *et al.*, 1987; Myers *et al.*, 1995). Cheng & Smith (1989) reported the distribution of Chi stimulated change events and extent of the heteroduplex region (Fig. 2.8). Also, the distribution starts at Chi indicating the final cleavage happening directly at Chi.

Bipolar DNA translocation contributes to highly processive DNA unwinding by RecBCD enzyme (Dillingham *et al.*, 2005).

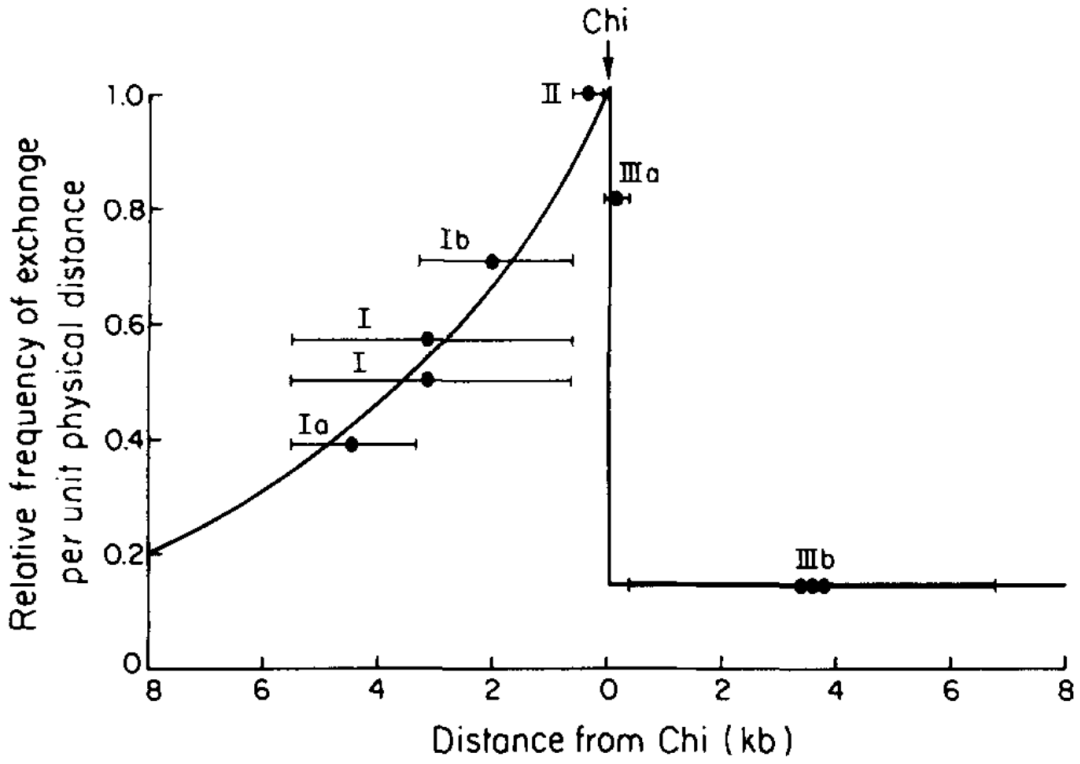


Figure 2.8. Distribution of recombinational exchanges in the presence of Chi (Cheng & Smith, 1989). Chi-stimulation is maximal near Chi and an exponentially decreasing distribution of exchanges to the left of Chi, decreasing 2-fold for each 3.2 kb.

2.1.7 Models of Chi-activity

Myers *et al.* (1995) have observed that Chi recombination exchange activity decays exponentially. The decrease in Chi activity is simply a function of the physical distance (in DNA base pairs) from Chi as if the Chi-activated enzyme was subjected to spontaneous dissociation from its substrate.

2.1.8 RecA filament formation

RecA polymerises on the DNA to form a protofilament that enables strand exchange as described above.

RecA polymerisation on ssDNA is a complex process. RecA loading occurs in two steps - nucleation or initial binding to the DNA (slow) and filament extension, e.g. addition of monomers (fast) (reviewed by Lovett (2012)). RecA also needs to displace SSB protein which is normally bound to ssDNA. The mechanism of RecA assembly on ssDNA is summarised in Fig. 2.9.

Filament formation *in vitro* is slow and a mediator protein is needed to efficiently compete with SSB protein for the binding place on the ssDNA. Some enzymes like RecF (Bell *et al.*, 2012) and RecBnuc (Arnold *et al.*, 2000; Spies & Kowalczykowski, 2006) promote only nucleation whereas RecOR helps both nucleation and filament extension (Bell *et al.*, 2012). RecO binds directly to SSB protein and has been suggested to help removing it from the DNA (Umezu & Kolodner, 1994). Spies & Kowalczykowski (2006) demonstrated the existence of a stable complex formed between RecBnuc (nuclease domain of RecB) and RecA and suggested that this acts as a catalyst of RecA binding to the DNA *in vivo*. By directly binding RecA, RecBnuc might increase the concentration of RecA, thus increasing the chances of formation of a nucleus.

RecA polymerisation on the ssDNA requires ATP binding but not ATP hydrolysis (Galletto *et al.*, 2006). Lovett (2012) suggested that ATP binding increases the affinity of RecA to ssDNA by inducing a change in the conformation of RecA.

Nucleation was shown to be faster on single-stranded DNA (ssDNA) than on double-stranded DNA (dsDNA) (Cox, 2007). The initiation of clustering of RecA

on ssDNA needs at least a dimer (Bell *et al.*, 2012). This is the smallest RecA oligomer that has the capacity to bind ATP (Chen *et al.*, 2008).

The preferred position of RecA dimer transfer by RecBnuc on the 3' single strand is unknown. The growth of the filament occurs preferentially in the $5' \rightarrow 3'$ direction with a rate of $120 - 1200$ $subunits/min$ (Galletto *et al.*, 2006; Joo *et al.*, 2006; Shivashankar *et al.*, 1999; Van Der Heijden *et al.*, 2005). The growth rate in the $3' \rightarrow 5'$ direction is about twice slower (Bell *et al.*, 2012).

RecA polymerisation is discontinuous (Churchill *et al.*, 1999) and one RecA covers three nucleotides. The monomers constantly dissociate and rebind to the filament which makes them highly dynamic structures (Lovett, 2012).

Dissociation of RecA monomers is dependent on hydrolysis of ATP (Galletto *et al.*, 2006). The monomers dislocate from ssDNA at a rate of ~ 70 monomers per minute (Arenson *et al.*, 1999) which is much slower than the rate of RecA polymerisation. *In vivo*, some proteins such as RecX and DniB regulate RecA filament formation. The RecX protein blocks the extension of RecA filaments during assembly (Drees *et al.*, 2004) and DniB prevents RecA dissociation. The mechanism of RecA assembly on the ssDNA coated with SSB protein (natural state of 3' ssDNA loop) has been studied *in vitro* using a single molecule approach (Bell *et al.*, 2012). Here are the key findings of the study:

- The filament assembly starts with a dimer nucleus.
- The number of clusters formed and the probability of nucleation increase linearly with time.
- Rate of nucleation:

$$J \propto k[RecA]^n \tag{2.1}$$

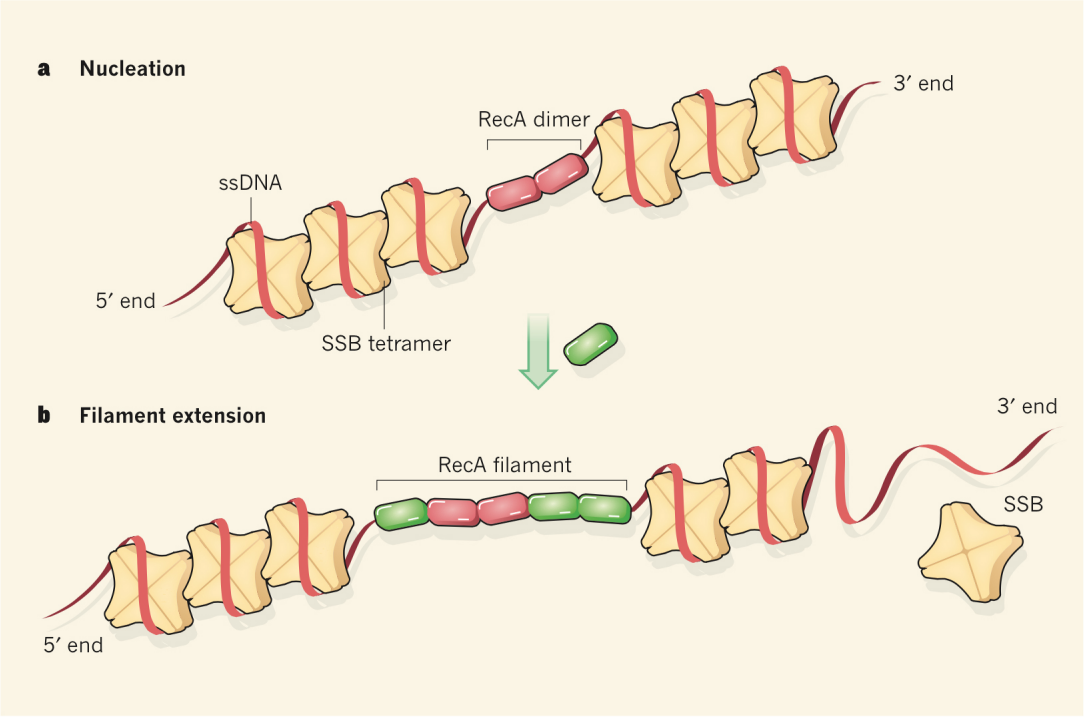


Figure 2.9. Mechanism of RecA filament assembly on the ssDNA, derived from Lovett (2012).

where $n \approx 2$.

- RecA stochastically forms multiple nuclei, which are subsequently extended in the growth phase.
- Increasing RecA concentration resulted in a net increase in the growth rate of individual clusters.
- RecA filaments grow via rapid addition of monomers.
- Growth rate is a linear function of the concentration of RecA supporting a monomeric addition type of growth model.
- The growth rate on SSB-coated DNA is about the same as in the bulk which suggests that SSB protein does not impede filament growth.

- Nucleation time is high at low RecA concentrations, but increases with concentration.
- RecA occupies its spot on a ssDNA upon SSB dissociation or sliding away.
- RecA filament growth on SSB-coated ssDNA is bidirectional.
- RecA competes with SSB protein for the binding sites on the ssDNA.
- One SSB molecule binds 65 nucleotides of ssDNA.

2.1.9 The role of RecBCD dual motor architecture

As previously mentioned, RecBCD employs its two helicases of different polarity to drive translocation and unwind the dsDNA before recognition of a Chi site. Prior to Chi recognition, RecD advances on the $5'$ strand taking a leading role in the unwinding process leaving RecB behind as a lagging helicase. Upon recognition of a Chi site and conformational change in RecBCD complex, RecD gets disengaged and the slower helicase remains to unwind the duplex DNA further on.

There have been many speculations in the literature as to the benefits of the dual motor architecture. Since the helicases are able to act independently even in the absence of another helicase and perform the same function on the DNA (unwinding of the duplex), the accidental inactivation of one of them should not disrupt the process of unwinding at least before recognition of a Chi site. In fact, RecBCD where both motors are active has been found to be significantly more processive than RecBCD with one of the two motors being inactive. Mutant RecBCD enzymes in which either of the two helicase motors is inactivated by mutagenesis showed not only reduced speed (by 30% for RecB and 50% for RecD) but also reduced processivity of translocation by approximately 25- and 6-fold for

RecD and RecB respectively (Dillingham *et al.*, 2005). The inactive motor does not lose its ability to bind ssDNA, so in this case it remains bound to the non-translocated strand close to the initiation site for unwinding (Taylor & Smith, 2003). The use of two DNA motors is potentially capable of generating more force than a single motor (Dillingham *et al.*, 2005).

Also, the theoretical study of Stukalin *et al.* (2005a) using stochastic discrete modelling explained how the interaction between the two coupling motors accelerates the speed of the complex, as compared with the velocities of the individual free moving domains.

2.1.10 Heterogeneity of RecBCD activity

Unwinding rates of each molecule tend to vary a lot (Bianco *et al.*, 2001; Handa *et al.*, 2005; Spies *et al.*, 2005, 2007). In their attempt to determine the source of this intrinsic heterogeneity a single molecule study by Liu *et al.* (2013a) established that unwinding of DNA by individual RecBCD molecules is bimodal and demonstrated the existence of two populations of RecBCD with different absolute speeds - fast (mean 1.5 kbp/s) and slow (mean 0.9 kbp/s) (Fig. 2.10). The bimodal distribution of the absolute speed was related to the variation in RecBCD helicase activity where RecBCD is trapped in one of a few distinct kinetic conformations. The switch between the two speed sub-states was suggested to manifest through conformational changes in the complex. This state transition however requires a trigger event, such as an induced arrest of the molecule. The probability of conformational transition was found proportional to the exponent of the pause length. The molecule arrest would lead to a destabilisation of the molecular folding structure and increase the probability

for the molecule to explore the energy landscape and eventually arrive at a new conformation.

Interestingly, the slow population coincided with the population where RecB or RecD is inactivated suggesting that the slow conformational state is the one where one of the two motors is disabled. This essentially suggests that RecBCD can spontaneously acquire the conformation where one of the motor is no longer active. Ligand binding was shown to lock the molecule in one of the conformational states. It would be legitimate to assume that the uncertainty in the kinetic conformation of RecBCD (since both substates are within reach at all times) can get resolved both ways at the time of binding, resulting in both motors being engaged or only one of the motors (RecB or RecD) being engaged. The fast molecules also showed higher processivity than the slow ones.

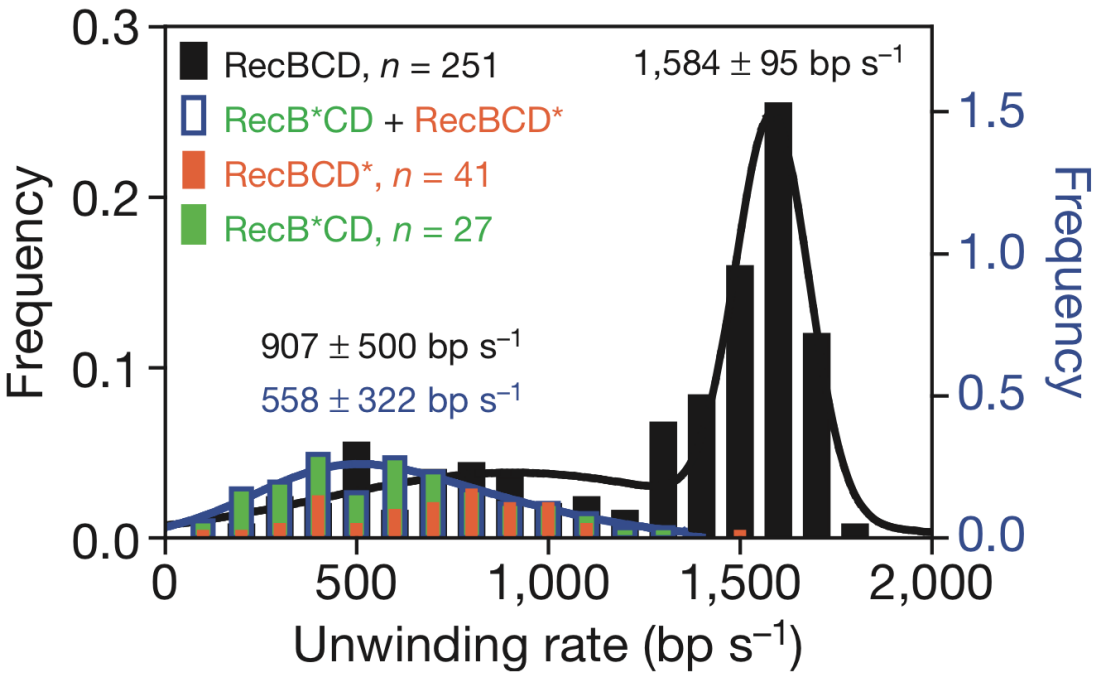


Figure 2.10. Distribution of unwinding rates for wild-type RecBCD and motor mutants, fit to the sum of two Gaussian functions and a single Gaussian, respectively (Liu *et al.*, 2013a).

2.1.11 A fixed point DSB

In this thesis I will use a system where a single site-specific DNA double-strand break is introduced into one copy of the replicated *E.coli* chromosome at a precise location. In order to produce a DSB in a fixed point of the DNA a palindrome has been inserted in *lacZ* gene (Eykelenboom *et al.*, 2008). The fork passage by this site leads to SbcCD cleavage of a DNA hairpin structure formed on only one of the replicated chromosomal copies and formation of a two side break (Fig. 2.11) on this copy of replicated DNA. Then presumably, the other intact chromosome (sister chromosome) is used as a template to repair the broken sister chromosome (Eykelenboom *et al.*, 2008).

2.2 ChIP-Seq

2.2.1 Introduction to ChIP-Seq

Chromatin immunoprecipitation followed by high-throughput sequencing (ChIP-Seq) is the standard methodology to map the chromosomal locations of DNA binding proteins such as transcription factors, DNA-binding enzymes, histones, chaperones, or nucleosomes (review by Bailey *et al.* (2013)). The steps of ChIP-Seq procedure are summarised on Fig. 2.13. First, the cells in a colony (containing $\sim 10^8$ cells) are fixed with formaldehyde in order to cross-link the protein of interest with the DNA at the location where it happened to bind at the moment of cell fixation. Next, the DNA is isolated and sheared into short fragments (150 - 500 nt) using ultrasound (sonication). The fragments are subjected to magnetic beads covered by the antibodies that selectively bind to the fragments carrying the protein of interest. Then immunoprecipitation follows where the magnetic

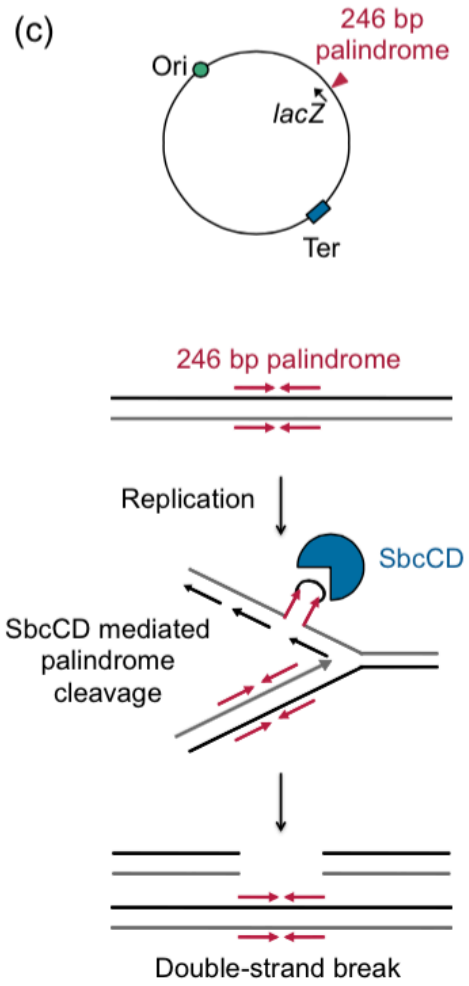


Figure 2.11. The hairpin endonuclease SbcCD is used to cleave a 246-bp interrupted palindrome inserted in the *lacZ* gene of the *E.coli* chromosome. Cleavage of this DNA hairpin results in the generation of a site-specific DSB on only one pair of replicating sister chromosomes, thus leaving an intact sister chromosome to serve as a template for repair by homologous recombination (from Cockram *et al.* (2015)).

Figure 1.1: Bioanalyzer traces of final library.

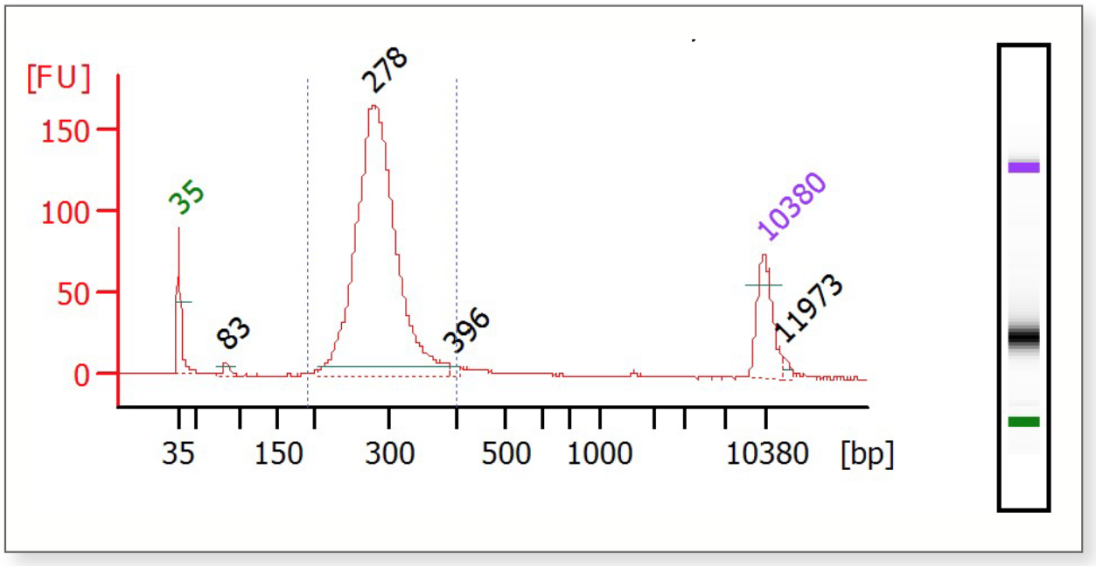


Figure 2.12. Bioanalyzer traces of final fragment library (after reverse cross-linking) prepared using NEBNext ChIP-Seq Library Prep Master Mix Set for Illumina. The graph shows the presence of fragments of various lengths in the library. The peaks correspond to the largest sub-populations of the fragments. Only 300 bp-fraction of the fragment library (separated from the rest with dotted lines) is selected for the next step in ChIP-Seq pipeline (sequencing). Reprinted from New England Biolabs, Inc. (Version 6.0).

beads carrying the antibodies bind to the protein bound DNA fragments, and then those beads are pulled down and the remaining fragments (not bound by the antibodies) are washed away. The following step is reverse cross-linking which allows to isolate pure protein-free DNA fragments constituting so called ChIP-Seq libraries. Afterwards, 300 nt long fragments are selected for PCR amplification (the fraction contained within dotted lines, Fig. 2.12)

After size selection and PCR amplification, all the resulting ChIP DNA fragments are sequenced simultaneously using the Genome Analyzer and Solexa Sequencing

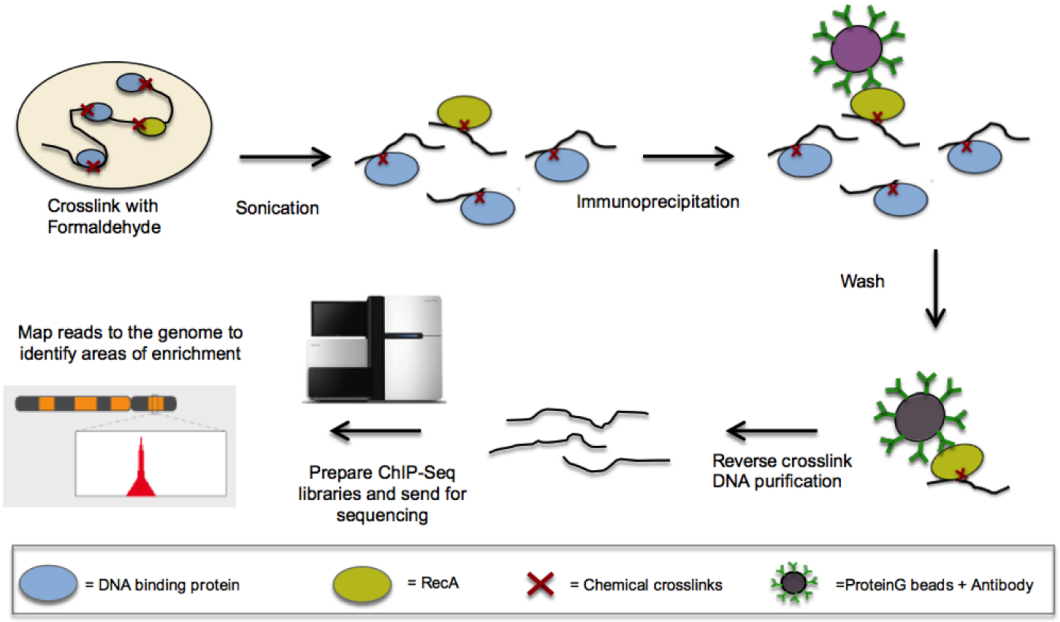


Figure 2.13. ChIP-Seq experiment, courtesy of Charlie Cockram.

technology. It is worth noting that when reading the base composition of fragments the sequencer may introduce a small per-base error which can be as low as $< 0.1\%$ provided a quality metrics is used for each base-call (Shendure & Ji, 2008). Since PCR amplification is not uniform across the genome (Goren *et al.*, 2010; Kozarewa *et al.*, 2009) identical reads are often removed (user-settable) in order to reduce sequence bias introduced by PCR amplification.

On the 5' end, the terminal 25-50 nt of the retained reads (later referred to as "tags") are then aligned to the reference genome. A successful matching event between a mapped read and the reference genome contributes hits to the fragment signal map (later referred to as "pileup data") one at each base pair along the stretch of the sequence aligned with a tag (Fig. 2.14). Often only uniquely mapped reads are retained. After all reads have been mapped the hits are pulled together to make pileup data which should be interpreted as distribution of fragment density across the reference genome. Fragment density

in turn should be translated into protein binding frequency. Mapping shorter tags increases the chance of multi-mapping (because of the regions of repeated DNA), yet longer tags may contain a sequencing error with a probability proportional to the length of the read. So, the length of the tag should be optimised as a trade-off between identifying a unique match and a correct one.

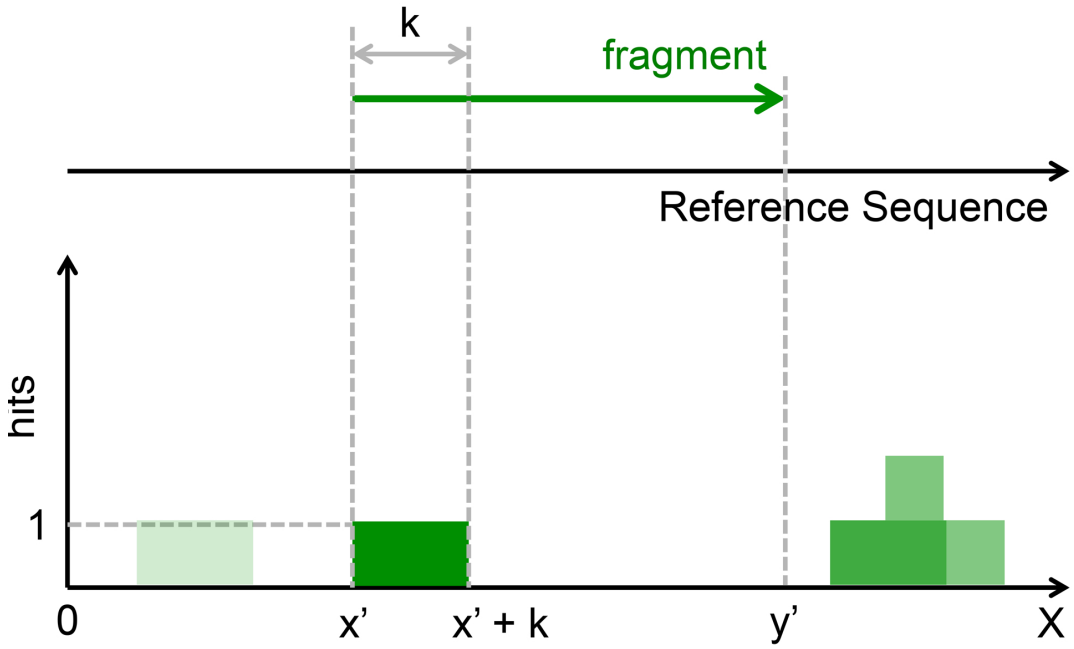


Figure 2.14. ChIP-Seq: mapping reads to the reference genome. Only k base pairs ($k = 25 - 50\ bp$) are aligned to the reference sequence. A hit is assigned to each genomic location within the matched stretch of the sequence.

2.2.2 Quantitative tools to analyse ChIP-Seq data

The purpose of ChIP-Seq analysis tools is to detect the regions of protein binding (areas of enrichment). These tools are called peak-calling algorithms, because they detect a peak at the locations where proteins occupy the DNA. There are a few quantitative tools available for the analysis of ChIP-Seq (review by Bailey

et al. (2013)). The most widely used are MACS (Zhang *et al.*, 2008a), PeakSeq (Rozowsky *et al.*, 2009a)), SAGE (Robinson & Smyth, 2007) and RNA-Seq (NB) (Robinson *et al.*, 2010). These tools do not compare the relative enrichments to assess the relative frequency of binding. Hence, these methods are quite restrictive in this respect because they aim to simply identify whether binding occurs or not.

2.2.3 Control sample

It is important to use an appropriate control data set prior to analysis of a ChIP-seq data set. This is because there are always sources of systematic bias present in the process of acquisition of ChIP-Seq data (Aird *et al.*, 2011; Li *et al.*, 2010). For instance, sonication of the DNA breaks it in an irregular manner (for a review see Landt *et al.* (2012)). Higher fragmentation of certain regions of DNA than others may lead to their overrepresentation in a ChIP-Seq pileup data set. In order to reliably identify the binding sites the data analysis pipeline has to include some reference signal which would be tested for potential bias prior to the analysis of a ChIP-Seq data set. DNA that has been processed under the same conditions as the immunoprecipitated DNA (called "Input" DNA) is generally used to produce such reference data set (for a review see Landt *et al.* (2012)). Contrary to immunoprecipitated DNA, fragments contained in the Input DNA enter the sequencing stage directly without passing through the selective antibody filter (no immunoprecipitation step, depicted in the right upper corner of Fig. 2.13). Zhang *et al.* (2008b) showed that the distribution of counts in the Input (pileup data set generated after sequencing Input DNA) is not simply uniform but is mildly fluctuating and contains some regions with relatively high ChIP enrichment comparable to some binding sites. The presence of such background spots with higher than average enrichment in the ChIP-Seq

data may be associated with the systematic bias introduced during the acquisition of ChIP-Seq data.

2.3 Summary of the common discrete probability models

Here I will summarise some common probability distribution models used in my thesis.

2.3.1 Geometric distribution

The probability distribution of the number of successes in a sequence of Bernoulli trials before a failure is reached is

$$Pr(X = k) = (1 - p)^{k-1}p,\ k = 1, 2... \tag{2.2}$$

$$\mathrm{E}(X) = \frac{1}{p} \tag{2.3}$$

$$\mathrm{Var}(X) = \frac{1 - p}{p^2} \tag{2.4}$$

2.3.2 Negative binomial distribution

In a sequence of Bernoulli trials, the probability of seeing k successes by the time r failures have been accumulated is

$$Pr(X = k) = \binom{k + r - 1}{k} p^k (1 - p)^r \tag{2.5}$$

$$\mathrm{E}(X) = \frac{pr}{1-p} \tag{2.6}$$

$$\mathrm{Var}(X) = \frac{pr}{(1-p)^2} \tag{2.7}$$

2.3.3 Multinomial distribution

After n draws with replacement from a pool of k different types of items where the probability of drawing an item that belongs to the k^{th} group is p_k, the probability of collecting a set of $n_1, n_2, ..., n_k$ items is described by the multinomial distribution (MN).

$$Pr(X_1 = n_1, X_2 = n_2, ..., X_k = n_k) = \binom{n}{n_1 n_2 ... n_k} p_1^{n_1} ... p_k^{n_k} \tag{2.8}$$

$$n = \sum_{i=1}^{k} n_i$$

$$\mathrm{E}(X_i) = np_i \tag{2.9}$$

$$\mathrm{Var}(X_i) = np_i(1-p_i) \tag{2.10}$$

2.4 Finite state discrete absorbing Markov Chain

Definition 1 (Markov Chain)**.** *Say we have a finite set of states* $X = \{1, 2, 3, ..., j, ..., K\}$.

A Markov Chain (MC) is a sequence of states $X_i \in \{X\} : (X_1, X_2, ..., X_n)$ sampled from this state space, which are visited stochastically according to some predefined transition rules between the states :

$$p_{ij} = p(X_{n+1} = j | X_n = i)$$

$$p_{ii} = p(X_{n+1} = i | X_n = i)$$

The main property of MC is that the probability of the next transition is solely dependent on the previous state and not on the history of the preceding transitions.

$$p_{ij} = p(X_{n+1} = j | X_n = i) = p(X_{n+1} = j | X_n = i, X_{n-1}, X_{n-2}, ..., X_1)$$

Definition 2 (Transient state)**.** *The state of the MC is called transient if the probability of leaving that state is nonzero*

$$\exists j \; p(X_{n+1} = j | X_n = X_{transient}) > 0$$

Definition 3 (Absorbing state)**.** *If a Markov chain reaches an absorbing state it resides there forever, the probability of leaving the absorbing state being zero.*

$$p(X_{n+1} = j | X_n = X_{abs}) = 0 \; if \; j \neq X_{abs}$$

$$p(X_{n+1} = j | X_n = X_{abs}) = 1 \; if \; j = X_{abs}$$

Definition 4 (Multidimensional MC)**.** *A multidimensional MC is a generalisation of MC where the state space is multidimensional, in the discrete case - cartesian product of one-dimensional discrete state subspaces:*

$$\Lambda = \{X\}^{(1)} \times \{X\}^{(2)} \times ... \times \{X\}^{(m)}$$

The element of this space $X \in \Lambda$ is a node on the m-dimensional grid.

Multidimensional (m-dimensional) MC process is a vector

$$\boldsymbol{X}_i = [X_i^{(1)}...X_i^{(m)}], i = 1...n$$

Definition 5 (MC Transition Matrix)**.**

$$T = \{p_{ij}\}, i, j \in \Lambda$$

The transition matrix contains all the transition probabilities between each pair of states i and j.

Definition 6 (Parametrically defined MC Transition Matrix)**.** *Parametric form of the transition probabilities:*

$$p_{ij} = p_{ij}(\theta)$$

Then the parameter form of the Transition matrix is:

$$T = T(\theta)$$

Theorem 1. *Let T be the transition matrix of a MC. After n transitions:*

$$T_{ij}^n = p_{ij}^{(n)}$$

The $(ij)^{th}$ element of this matrix is a probability of transfer from state i to state j after n steps.

Definition 7 (MC's initial probability distribution)**.** *Initial state vector $\boldsymbol{\pi}_0$ represents the probability distribution of occupying each state in the beginning of a process.*

$$\sum \boldsymbol{\pi}_0 = 1$$

Theorem 2. *The state vector after n steps given the initial state vector $\boldsymbol{\pi}_0$ and*

the transition matrix T is

$$\boldsymbol{\pi}_n = \boldsymbol{\pi}_0 T^n$$

Definition 8 (MC's stationary probability distribution)**.** *The stationary probability distribution is defined as:*

$$\boldsymbol{\pi}_\infty = \lim_{n\to\infty} \boldsymbol{\pi}_0 T^n = \boldsymbol{\pi}_0 T^\infty$$

Theorem 3 (Convergence to the absorbing subspace)**.** *If the process terminates by getting trapped in one of the absorbing states, the state vector will converge to the absorbing state subspace*

$$\boldsymbol{\pi}_\infty = \boldsymbol{\pi}_0 T^\infty = \boldsymbol{\pi}_{abs}$$

where all the elements of this state vector not corresponding to the absorbing states are equal to zero.

2.5 Parameter inference from the data

2.5.1 Maximum log-likelihood (MLE)

Suppose, we have a model with unknown parameters θ, $x \sim p(x;\theta)$.
For a data vector of independent measurements of identically distributed random variable x, $\mathbf{x} = x_1, ...x_n$, the likelihood of the data is

$$L(\mathbf{x};\theta) = \prod_{i=1}^{n} p(x_i;\theta)$$

Choose θ that increases the likelihood of the data to occur

$$\hat{\theta} = \arg\max_{\theta} p(\mathbf{x};\theta)$$

Sometimes it is easier to maximise the logarithm of the likelihood $\mathcal{L}(\mathbf{x};\theta) = \ln p(\mathbf{x};\theta)$ instead of the likelihood itself.

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(\mathbf{x};\theta)$$

$$\mathcal{L}(\hat{\theta}) \geq \mathcal{L}(\theta),\ \forall\theta$$

$1-\alpha$ confidence interval for θ (Θ^-, Θ^+) is

$$P(\Theta^- \leq \theta \leq \Theta^+) > 1-\alpha,\ \forall\theta$$

2.5.2 Maximum log-likelihood (MLE) of a discrete distribution with truncated support

Suppose the data ($\mathbf{n}$) consists of N independent random variables x_i drawn from k independent groups, so that there are n_i observations in the i^{th} group for each $i = 1, ..., k$.

$$\sum_{i=1}^{k} n_i = N$$

$\pi(i;\theta)$ is the probability of a randomly chosen observation to belong to the i^{th} group.

$$\sum_{i=1}^{k} \pi(i;\theta) = 1$$

The number of observations over k groups follows a multinomial distribution since the measurements are independent.

The likelihood of the data ($\boldsymbol{n} = (n_1, ..., n_k)$) given the parameter set θ is

$$L(\boldsymbol{n}; \theta) = N! \prod_{i=1}^{k} \frac{\pi(i; \theta)^{n_i}}{n_i!}$$

The log-likelihood is

$$\mathcal{L}(\boldsymbol{n}; \theta) = \sum_{i=1}^{k} n_i \ln \pi(i; \theta) + \ln N! - \sum_{i=1}^{k} \ln n_i!$$

The Maximum-Likelihood Estimate of θ (MLE) is

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(\boldsymbol{n}; \theta) \Rightarrow \frac{\partial \mathcal{L}}{\partial \theta}\Big|_{\hat{\theta}} = 0$$

Parameter independent MLE

After eliminating the parameter dependency of π on θ, we need to solve the following problem:
Maximise

$$\mathcal{L}(\boldsymbol{\pi}) = \mathcal{L}(\boldsymbol{n}; \theta) = \sum_{i=1}^{k} n_i \ln \pi(i; \theta) + \ln N! - \sum_{i=1}^{k} \ln n_i!$$

subject to the constraint

$$g(\boldsymbol{\pi}) = \sum_{i} \pi_i = 1$$

Since $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\pi}} = 0$ at its maximum and g is a constant, by introducing Lagrange multiplier γ find π which maximises $\mathcal{L}$ by solving the following system of equations

$$\forall i,\ 1 \leq i \leq k,\ \frac{\partial \mathcal{L}}{\partial \pi_i} - \gamma \frac{\partial g}{\partial \pi_i} = 0 \Leftrightarrow$$

$$\frac{n_i}{\pi_i} - \gamma = 0 \Rightarrow$$

$$\sum_i (n_i = \gamma \pi_i) \Leftrightarrow$$

$$\Rightarrow \pi_i = \frac{n_i}{N}$$

Plugging the latter into the expression for the log-likelihood we obtain

$$\max \mathcal{L}(\boldsymbol{\pi}) = \sum_{i=1}^{k} n_i \ln \frac{n_i}{N} + C \tag{2.11}$$

where C is a constant independent of $\boldsymbol{n}$.

Eq. 2.11 gives the absolute maximum of the likelihood function that can be possibly reached given data $\boldsymbol{n}$. This means that any likelihood function of the same data but parametrically constrained ($\theta \in \Theta$) will be smaller than or equal to the parameter-free likelihood which depends only on the data.

$$\sup_{\theta \in \Theta} \mathcal{L}(\boldsymbol{n}; \theta) = \sum_{i=1}^{k} n_i \ln \frac{n_i}{N} + C \tag{2.12}$$

2.5.3 Likelihood ratio test (LRT)

The likelihood ratio test compares two hypotheses:
$H_0 : \theta \in \Theta_0 \subset \Theta$ against $H_1 : \theta \in \Theta$.

Definition 9 (Likelihood ratio Statistic)**.** *Log-likelihood ratio (LR) statistic is*

$$\lambda(\boldsymbol{n}) = 2[\sup_{\theta \in \Theta} \mathcal{L}(\boldsymbol{n}; \theta) - \sup_{\theta \in \Theta_0} \mathcal{L}(\boldsymbol{n}; \theta)] \tag{2.13}$$

LRT requires to reject H_0 with probability $\alpha(c)$, where $c = \lambda(\boldsymbol{n})$ & $c \in [0, 1]$.

Theorem 4 (Asymptotic distribution of LR)**.** *LR statistic converges in distribution to Chi-square statistic*

$$\lambda(\boldsymbol{n}) \xrightarrow{d} \chi^2_p$$

with degree of freedom

$$p = \dim \Theta - \dim \Theta_0$$

in the limit of large sample size $n \to \infty$

This means

$$P(\lambda(\boldsymbol{n}) \geq c) = P(\chi_p^2 \geq c) = \alpha(c)$$

So, if $\lambda \geq c$ this can happen by chance only with probability $\alpha(c)$.
If $\alpha(c)$ is low, say below some low probability threshold, for example $\alpha(c) < 5\%$, H_0 can be rejected, because it is extremely unlikely to occur, the chance of that being under 5%.

2.5.4 Goodness-of-fit using LRT

Having substituted Eq. 2.12 into Eq. 2.13, LR statistic to test $H_0 : \theta \in \Theta_0 \subset \Theta$ against $H_1 : \theta \in \Theta$ is

$$\lambda = 2 \sum_{i=1}^{k} n_i \ln \Big(\frac{n_i}{N \pi_i(\hat{\theta})} \Big)$$

Degrees of freedom

$$\sum_{i=1}^{n} \pi = 1 \Rightarrow \dim \Theta = k - 1$$

$$\dim \Theta_0 = p \ (p = length(\theta))$$

Then according to the LR-theorem

$$\lambda \xrightarrow{d} \chi_{k-p-1}^2$$

This means

$$\alpha(c) = P(\lambda \geq c) = P(\chi_{k-p-1}^2 \geq c)$$

Hence, if $\lambda > c(\alpha)$ we reject $H_0 : \theta \in \Theta_0$ in favour of $H_1 : \theta \in \Theta$ with probability α.

On the contrary if H_0 was accepted at α-level this would mean that the $\hat{\theta}$-model fits data $\boldsymbol{n}$ with $(1-\alpha)100\%$ confidence.

2.5.5 Asymptotic confidence intervals

The LR testing principle can also be used to construct confidence intervals.

The idea is to find all the values of parameter θ for which the log-likelihood $\mathcal{L}(\boldsymbol{n};\theta)$ does not differ too much from the maximum log-likelihood $\mathcal{L}(\boldsymbol{n};\hat{\theta})$ or the log-likelihood statistic does not exceed a chosen cut-off $c(\alpha)$. Here, we test the null hypothesis $H_0 : \theta = \theta_0$ vs. its alternative $H_1 : \theta \neq \theta_0$.

LR statistic is

$$\lambda(\boldsymbol{n}) = 2(\mathcal{L}(\boldsymbol{n};\hat{\theta}) - \mathcal{L}(\boldsymbol{n};\theta_0)) \tag{2.14}$$

If $\lambda(\boldsymbol{n}) > \chi^2_{p,\alpha}$ we would reject H_0 at α-level.

Alternatively, an approximate $100(1-\alpha)\%$ confidence interval for θ when the sample size n is large would consist of all the possible θ_0s for which the hypothesis $H_0 : \theta = \theta_0$ would not be rejected at the α level (see Asymptotic result for the likelihood ratio (LR) statistic).

For example, suppose the cut-off probability is 5% for rejection and Θ is a one-dimensional parameter space ($p = 1$) then

$$\chi^2_{p,\alpha} = \chi^2_{1,0.05} = 3.8$$

and Eq. 2.14 becomes

$$2(\mathcal{L}(\boldsymbol{n};\hat{\theta}) - \mathcal{L}(\boldsymbol{n};\theta_0)) \leq 3.8 \tag{2.15}$$

So, the confidence interval for $\hat{\theta}$ consists of all those θ_0 that satisfy the inequality in Eq. 2.15.

2.6 Model optimisation techniques

2.6.1 Local minimum by Gradient Descent

Gradient Descent requires differentiability of a function $F(\theta)$ at the point where it reaches its maximum a. $F(\theta)$ should be differentiable in the vicinity of a.
The first order Taylor approximation of $F(\theta)$

$$F(\theta) = F(\theta_0) + \nabla F(\theta_0)(\theta - \theta_0) + O(||\theta - \theta_0||^2)$$

Let

$$\theta = \theta_0 + hu$$

where u is a unit vector.

$$F(\theta_0 + u) - F(\theta_0) = h\nabla F(\theta_0)u + h^2 O(1)$$

In order to decrease $F(\theta_0 + u)$, $\nabla F(\theta_0)u$ needs to be minimised

$$min \nabla F(\theta_0)u = -\nabla F(\theta_0)/||\nabla F(\theta_0)||$$

Algorithm

Initial conditions:

- initial guess θ_0
- maximum number of interactions N_{max}
- gradient norm tolerance ϵ_g
- step tolerance ϵ_θ

$$F(\theta_0) \geq F(\theta_1) \geq ...$$

If a local minimum exists the sequence will eventually converge to it unless the maximum number of steps has been reached.
The step size β_i must be chosen to increase the convergence speed and prevent divergence. β_i should not be too large because then the first order approximation would become invalid. On the other hand very small step sizes would lead to slow convergence.

2.6.2 Global minimum by Grid Sampling

Let us define a continuous function $F(\mathbf{x}) : [0,1]^n \to \mathbb{R}$.
F($\mathbf{x}$) may have several local minima, so the Gradient Descent algorithm to identify the global minimum is not reliable in this circumstance.
We need to estimate approximate $\mathbf{x}_{min}$ where the function is close to its global minimum, so that this value can be used as a first guess for the Gradient descent algorithm to estimate the global minimum with higher precision.
Those are the steps to estimate a global minimum with grid sampling:

- Discretise the support of the function $[0,1]^n$ to estimate the minimum of the function with a given precision. $\epsilon_i, i = 1, ..., n$ are the discretisation steps for each dimension.

- Compute the value of the function at each point of the discretised space.
- Choose the argument value where the function reaches its minimum.

This algorithm can be practically used provided:

- the number of dimensions is small, ideally ($n \leq 3$),
- the function is expected to be monotonic between each two adjacent points of the discretised space
- the discretisation step is not too small
- the computation of the function value at one point does not take too much time

Grid sampling restricts the precision with which the minimum of the function can be estimated. Suppose we need to estimate the value of the minimum of the three-dimensional function ($n = 3$) with a precision of at least 0.05. In order to do that we have to perform 20^3 computations. If the time required to evaluate the function at one point is $1s$, it takes $20^3 s$ ($2.2h$) of computational time to identify the point where the function reaches its minimum.

In order to improve the precision of the initial guess to feed into the Gradient Descent algorithm (and by doing so improve the chances of converging to the global minimum), sequential grid sampling can be performed, whereby the same algorithm outlined above is run several times, each time focusing on the estimate obtained in the preceding cycle of a more coarse-grained algorithm and refining the grid around it to reestimate the minimum with higher precision.

2.6.3 MLE by Grid Sampling

Suppose we need to identify the maximum of a log-likelihood function $\mathcal{L}(\boldsymbol{n};\theta)$, where $\theta \in \Theta = [0,1]^n$ in order to determine the MLE estimate $\hat{\theta}$ and its confidence interval. However, the complexity of the function does not allow us to find the maximum analytically and we need to use numerical methods to identify these parameters.

Also, we do not know how many local maxima of $\mathcal{L}(\boldsymbol{n};\theta)$ exist on its support, which means we cannot use a Gradient Descent algorithm for this purpose straightaway, because this method would converge to a local maximum and may miss the global maximum depending on the initial conditions.

If the number of parameters is $n \leq 3$ we can employ a Grid Sampling method at least to arrive at an initial guess which will be fed into a Gradient Descent algorithm, or simply report θ with a precision equal to the interval used for Grid Sampling.

Prior knowledge about the possible location of the optimal value of the parameter can also be helpful when using a Grid Sampling algorithm to reduce the computational time.

Here, for simplicity I will consider a one-dimensional case $\Theta = [0,1]$. Suppose, somehow we acquired more detailed information about the constraints on parameter $\theta : \theta \in \Theta^* = [\theta_-, \theta_+]$. The task now is to identify the most likely interval of θ.

The next step is to discretise Θ^* space with an interval ϵ which we choose optimal

to balance the desired precision and the computational cost.

$$\Theta^* = \{\theta_-, \theta_- + \epsilon, \theta_- + 2\epsilon..., \theta_+ - \epsilon, \theta_+\} = \{\theta_1, \theta_2, ..., \theta_n\}$$

Then, compute the likelihood function at each point of the discretised parameter space

$$L(i) = L(\boldsymbol{n}; \theta_i)$$

Now, find the maximum of L and report the estimate of θ corresponding to the maximum of L

$$\hat{\theta} = \theta_i = \arg\max_i \mathcal{L}(\boldsymbol{n}; \theta_i)$$

Then, report the confidence interval of θ using Eq. 2.15

$$\hat{\Theta} = \{\theta;\ 2(\mathcal{L}(\boldsymbol{n}|\hat{\theta}) - \mathcal{L}(\boldsymbol{n}|\theta)) \leq 3.8 \wedge \theta \in \Theta^*\}$$

2.6.4 Metropolis-Hastings algorithm

Let $X \sim p(x)$ be an unknown probability distribution of x.
If we cannot compute this distribution analytically for certain reasons such as, when $p(x)$ requires the computation of a high dimensional integral for the normalisation constant, for example, then we need to approximate it numerically.

If $p(x)$ can be evaluated up to a proportionality constant $f(x) = p(x) * c$ then we can use Metropolis-Hastings algorithm (Hastings, 1970; Metropolis *et al.*, 1953).

Theorem 5 (Metropolis-Hastings)**.** *The sequence of selected x_j obtained by Algorithm 1 converges to x in distribution*

$$x_j \xrightarrow{d} X,\ X \sim f(x)$$

Data: $f(x)$, N - number of iterations, x_0 - initial guess, $q(x_{j+1}|x_j)$ - proposal kernel

Result: $hist(x_j), j = 1, ..., N$

for $j \in N$ **do**

Propose the next step $x^* \sim q(x^*|x_j)$

Calculate the acceptance probability:

$$\mathcal{A}(x_j|x^*) = \min\left\{1, \frac{f(x^*)}{f(x_j)}\frac{q(x_j|x^*)}{q(x^*|x_j)}\right\}$$

Choose a probability - a random number uniformly distributed on $[0, 1]$

$$u \sim \mathcal{U}_{[0,1]}$$

if $u < \mathcal{A}$ **then**

$x_{j+1} = x^*$;

else

$x_{j+1} = x_j$;

end

end

Algorithm 1: Metropolis-Hastings algorithm

Definition 10 (Symmetric random-walk Metropolis algorithm (RWM))**.** *For RWM the proposal distribution is symmetric*

$$q(x_j|x^*) = q(x^*|x_j)$$

which leads to

$$\mathcal{A}(x_j|x^*) = \min\left\{1, \frac{f(x^*)}{f(x_j)}\frac{q(x_j|x^*)}{q(x^*|x_j)}\right\} = \min\left\{1, \frac{f(x^*)}{f(x_j)}\right\}$$

The choice of a proposal distribution for a particular target distribution is a central problem in the application of RWM algorithm. The simplest proposal distribution is a uniform distribution on $[-\epsilon, \epsilon]$ - $\mathcal{U}_{[-\epsilon,\epsilon]}$ where ϵ - scaling factor of RWM - should be chosen such as to maximise the efficiency of the algorithm.

$$q(x^*|x) = \mathcal{U}_{[-\epsilon,\epsilon]}$$

The efficiency of a M-H algorithm depends on the scaling of the proposal density. Large scaling factors would lead to large variances of the proposal distribution $q(x^*|x)$, and that in turn would increase the rejection rate, so the random walker would get trapped in the same position for a long period of time with only a low chance of escaping that state. On the other hand a small variance would force the algorithm to accept the proposed steps and a very large number of steps would be necessary to explore the probability space, because it is rejection rather than acceptance that allows us determine the shape of the target distribution. This problem has been recognised early by Metropolis *et al.* (1953).

Thus, the optimal scale ϵ should be chosen to find the balance between the two extremes. Sometimes the optimal scale ϵ can be determined empirically, by trial and error, to achieve an acceptance rate which is far from 0 and far from 1. In recent years a more sophisticated method to find an optimal scaling utilises the principle of machine learning (so called adaptive MCMC) where the program learns the optimal parameters as it runs (for a review see Rosenthal *et al.* (2011)). Another useful criterion to consider is the number of iterations necessary to explore the target distribution. If a characteristic size of the distribution is L and we use the kernel with scale σ, then according to the property of a random walk the number of iterations to move all the way along the whole span of the distribution is

$$N \geq (L/\sigma)^2$$

This number should obviously scale with the dimension of the parameter space as $N \sim \sqrt{d}$.
For example if $L \sim 10\sigma$, we would need $N = 100$ iterations to explore the whole span of the distribution.

Suppose we know nothing about the target distribution except for the fact that the parameter is constrained within an interval, say $\theta \in [0, 1]$. Then the span of the distribution cannot exceed 1 and a choice of σ around 0.01 would give us the upper bound on the number of iterations

$$N_{min} < 10^4$$

2.7 Model selection using the Bayesian Information Criterion (BIC)

The Bayesian Information Criterion was introduced by Schwarz *et al.* (1978). BIC is an asymptotic approximation to a transformation of the Bayesian posterior probability of a chosen model. BIC of a model depends on the likelihood of the data given the parameters θ of the model $L(D|\theta)$. BIC is used to select one model over another, the model with significantly higher BIC should be rejected.

$$BIC = -2 \ln \mathcal{L}(\boldsymbol{n}; \theta) + k \ln n. \tag{2.16}$$

k is the number of parameters and n is the total number of measurements in the data $\boldsymbol{n}$

$$n = \sum_i n_i$$

and $k \ln n$ is the penalty term for using more parameters.

To test a model (*model* 2) with parameters θ_2 against another model (*model* 1) with parameters θ_1 on the same data $\boldsymbol{n}$ find the difference in BIC scores

$$BIC_2 - BIC_1 = 2 \ln \mathcal{L}(\boldsymbol{n}|\theta_1) - 2 \ln \mathcal{L}(\boldsymbol{n}|\theta_2) + \Delta k \ln n. \tag{2.17}$$

where $\Delta k = k_2 - k_1$ is the difference in the number of parameters between the models.

If the difference is positive and significantly large we say there is enough evidence to prefer *model* 1 over *model* 2, otherwise both models are considered statistically equivalent.

$BIC_2 - BIC_1$	Evidence Against *model* 2
$0 - 2$	Not worth more than a bare mention
$2 - 6$	Positive
$6 - 10$	Strong
> 10	Very Strong

Chapter 3

Markov Chain model of a sequence-switchable stochastic machine

3.1 Sequence-switchable stochastic machine

First of all let us introduce a particular case of a discrete-time finite state stochastic machine (SM). It is able to discriminate elements by scanning them and to switch its *internal state* in response to particular sequence elements, herein referred to as SWITCHes. This framework will be later used as a computational model for RecBCD activity on the DNA - an example of DNA processing enzymes.

Let the following be the properties of the sequence-switchable SM:

- The sequence Σ_1 is composed of elements σ that are randomly picked from some alphabet $\sigma \in \Sigma$; Σ also contains special elements, referred to as

SWITCHes $\sigma^* \in \Sigma$; the sequence of elements $\Sigma_1 = \{\sigma_1, \sigma_2, ..., \sigma_X\}$ serves as an input to SM.

- SM translocates in a probabilistic manner along the sequence using its two motors - fast (A) and slow (B); both motors are independent and step in parallel; they can only transition between the adjacent elements of the sequence.

- Here we restrict the movements of the motors to only one direction but the model can easily be extended to account for the case where stepping in the opposite direction is also allowed.

- It is the slow motor B that is allowed to perform the "sequence discrimination" computation by scanning it for the presence of SWITCH sites (σ^*). The current element of the sequence scanned by SM when motor B is at x is

$$\sigma = \sigma(x), x \in \mathfrak{X} = (1, ..., X)$$

- When arriving at a SWITCH ($x = x^*$) motor B responds to it in a probabilistic manner by changing the *internal state* of the SM

$$s_0(\sigma^*) \xrightarrow{recognition} \{s_0 \mapsto p(no\ switch), s_{switch} \mapsto p(switch)\}$$

where s_0 is the initial state of SM.

- As a result of the *internal state* transition of SM we request that motor B gets inactivated, e.g. the scanning is no longer allowed after SM's transition into the "switch" state s_{switch}; meanwhile motor A is allowed to continue propagating along the sequence until it stops spontaneously where SM reaches its terminals state s_{term}.

- The termination of SM's life cycle on the sequence is an internal decision of

SM independent of the environment; we only request that it always follows SWITCH-triggered transitions

$$s_{switch} \xrightarrow{termination} \{s_{switch} \mapsto p(no\ stop), s_{term} \mapsto p(stop)\}$$

where s_{term} is the terminal state of SM.

- SM can live infinitely in the transient state on the sequence provided the scanning motor B can also move backwards; in our case it will always terminate spontaneously with probability $p(stop)$ or when A arrives at the last element of the sequence $y = X$, y being the coordinate of A.

- To summarise the set of *internal states* of the SM which was defined above

$$Q = \{s_0, s_{switch}, s_{term}\} \tag{3.1}$$

- The transition relation of SM is

$$\Delta_{\mathcal{A}} \subset Q \times \mathcal{A} \times Q$$

where $\mathcal{A}$ is a set of actions

$$\mathcal{A} = \{transition, recognition, termination\}$$

- The phase diagram of the process of the SM interacting with a sequence is shown on Fig. 3.1.

- Define the probability function on the set of actions and inputs

$$P : \Delta_{\mathcal{A}} \times \Sigma \to [0, 1]$$

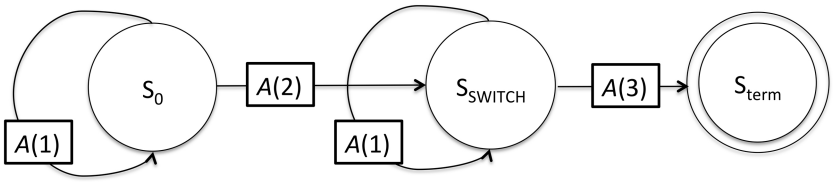


Figure 3.1. Phase Diagram of SM. $s_0, s_{switch}, s_{term}$ are the states of the SM. $\mathcal{A}$ is the set of actions performed by the SM on a sequence $\mathcal{A} = \{transition, recognition, termination\}$. $\mathcal{A}(1) - transition$, $\mathcal{A}(2) - recognition$, $\mathcal{A}(3) - termination$

$$P(s_0|s_0,\sigma) = \begin{cases} 1, & \text{if } \sigma \neq \sigma^* \\ 1 - p(switch), & \text{if } \sigma = \sigma^* \end{cases}$$

$$P(s_{switch}|s_0,\sigma) = \begin{cases} 0, & \text{if } \sigma \neq \sigma^* \\ p(switch), & \text{if } \sigma = \sigma^* \end{cases}$$

$$P(s_{switch}|s_{switch},\sigma) = 1 - p(stop)$$

$$P(s_{term}|s,\sigma) = \begin{cases} 0, & \text{if } s = s_0 \\ p(stop), & \text{if } s = s_{switch} \end{cases}$$

- Though the *internal states* of SM - Q - defined by its interaction with the sequence are hidden, what can be observed is the *trace* of the SM - i.e. the coordinates of the two motors A & B on the sequence. This *trace* is the externally visible output of the computation of SM on the sequence.

- Let us define *external states* of SM as the pair of locations of motors A and B on the sequence - the start and end positions of the trace of SM *trace* (x, y).

$$\epsilon = (x, y) \in \mathcal{X} \times \mathcal{X} \tag{3.2}$$

Finally let us extend the state space of SM by merging its *internal* (Eq. 3.1) and *external states* (Eq. 3.2). After having included the positions of the two motors on the sequence, the state space of SM is now a set of all possible combinations of *internal* and *external states*

$$(\epsilon, s) = (x, y, s) \in \mathcal{X} \times \mathcal{X} \times Q \tag{3.3}$$

3.2 Derivation of a Markov Chain model

In this section a discrete time finite state Markov Chain will be derived for the evolution of sequence-switchable SM.

The mechanical scanning of the elements of the sequence by SM and its decision making can be mathematically represented as a chain of memoryless transitions between the states of SM in the respective 3D state space (Eq. 3.3) where the probabilities of transitions between the states depend only on the current state of SM (x, y, s) in accordance with Markov property (Chapter 2).

The terminal state s_{term} of SM corresponds to an absorbing state of the MC $(x, y, s = 2)$, where $x \in \mathcal{X}$ - the start position of SM on the sequence - and $y \in \mathcal{X}$ - its end position - are random variables accumulated throughout the preceding chain of transitions. The probability distribution over the values of these random variables (*external state* of SM) - $p(x, y|s_{term})$ - is the central question posed in this chapter. This probability distribution should also be understood as a probability distribution over the absorbing state space of MC. This probability distribution will be utilised as a prior model in the construction of a statistical framework to analyse High Resolution Sequencing Data (HRSD).

3.2.1 MC transition rules

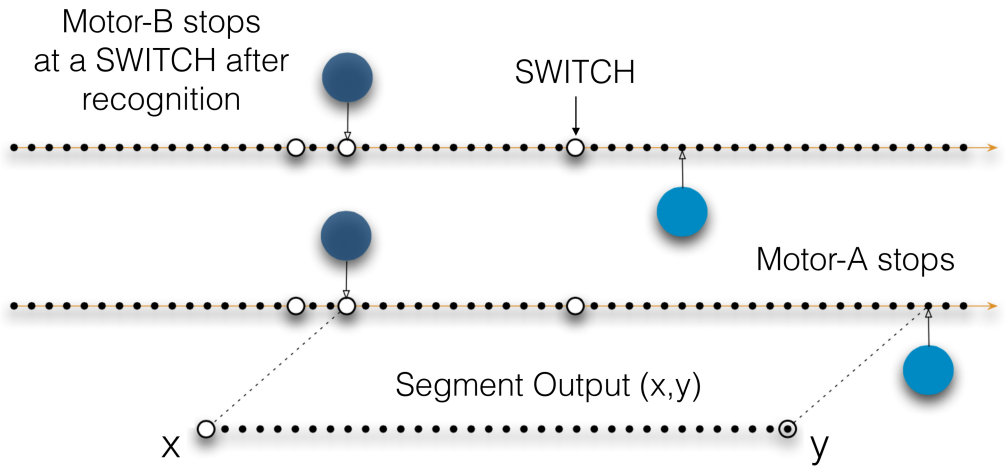


Figure 3.2. First, both motors propagate in parallel along the sequence. The slow motor (B) stops at some SWITCH site x^* (with probability p_χ). Afterwards, the fast motor (A) stops anywhere on the sequence after motor-B (with probability p_s) which leads to a full stop. The output generated by MC in its absorbing state is a random sequence interval (x, y) (created by Vincent Danos).

The progression of SM along the sequence starts with loading at some fixed position (we shift the coordinates of motors A and B with respect to the initial position of the machine ($x \to x - x_0, y \to y - y_0$). Then, upon initiation the *internal state* of SM is $s = s_0$ and the global state of SM is $(0, 0, s_0)$. The position of loading determines the sequence input for the SM (Σ_1).

Initially, SM proceeds through a sequence of transient states ($s_0 \to s_0$) scanning each element of the sequence using its motor-B, $\sigma(x), x \in \{x_0, ..., X\}$. Upon successful recognition of a SWITCH ($\sigma(x) = \sigma^*$), motor-B terminates there ($x = x^*$) driving MC into its absorbing subspace $S \to (x^*, y, s_{switch})$. In this state, the x-coordinate is no longer allowed to change. Yet in this "switch" state of SM, motor-A is still allowed to propagate according to the definition above

until the *internal state* of SM collapses into its terminal state ($s \to s_{term}$). This corresponds to the chain transition into its ultimate absorbing state $(x, y, s) \to (x^*, y^*, s_{term})$:

$$S_{abs} = (x^*, y^*, s_{term})$$

Now, we translate the assumptions about the sequence-switchable SM made above into MC transitions between any two states with non-zero probability. The transitions between any other two states are deemed impossible.

By construction, each step of MC leads to advancement of fast motor A, $y \xrightarrow{1} y{+}1$, with probability one.

Pre-recognition steps - the fast motor (A) is always ahead, the slow motor either makes a step simultaneously with A or skips a step during an MC step:

$$(x, y, s_0) \xrightarrow{p_+} (x, y+1, s_0)\ only\ A\ advances$$

$$(x, y, s_0) \xrightarrow{p_-} (x+1, y+1, s_0)\ both\ motors\ advance$$

Recognition step

$$(x, y, s_0) \xrightarrow{p_\chi} (x, y+1, s_{switch})\ Recognition\ success;\ x \in X^*$$

$$(x, y, s_0) \xrightarrow{q_\chi p_+} (x, y+1, s_0)\ Recognition\ failure\ and\ A\ advances;\ x \in X^*$$

$$(x, y, s_0) \xrightarrow{q_\chi p_-} (x, y{+}1, s_0)\ Recognition\ failure\ and\ both\ motors\ advance;\ x \in X^*$$

Post-recognition steps

$$(x, y, s_{switch}) \xrightarrow{q_s} (x, y+1, s_{switch})\ A\ advances$$

$$(x, y, s_{switch}) \xrightarrow{p_s} (x, y+1, s_{term})\ Stop,\ a\ termination\ state\ reached$$

Note that the process (Fig. 3.2) is monotonically increasing (on all 3 coordinates).

Also, I define $\theta \coloneqq [p_s, p_+, p_\chi]$ - the parameter vector governing the transitions between the states of MC.

3.3 Derivation of the probability distribution over the absorbing states

The distribution over the *absorbing external states* of the MC, $p(x^*, y^*|s_{term})$, can be computed given the initial state $(0, 0, s_0)$, the locations of SWITCH motifs X^* and the transition probabilities between the states are summarised in θ.

The derivation of $p(x^*, y^*|s_{term}; \theta)$ easily follows from the transition rules of MC listed above. Let us break down the number of steps made by motor-A before termination into:
y_1 - the number of steps made by motor-A in the initial state s_0;
y_2 - the number of steps made by motor-A in the "switch" state s_{switch};
The total number of steps made by A until termination is

$$y^* = y_1 + y_2 \tag{3.4}$$

In accordance with Markov property:

$$p(x^*, y_1, y_2|s_{term}; \theta) = p(x^*, y_2|s_{term}; \theta)p(x^*, y_1, s_{switch}; \theta)$$

Because x^* is fixed when the *internal state* is a "switch" state (s_{switch}), the above

relation can be simplified as

$$p(x^*, y_1, y_2|s_{term}; \theta) = p(y_2|s_{term}; \theta)p(x^*, y_1, s_{switch}; \theta)$$

Through parameter p_- (probability of motor-B advancing one sequence element in the initial state (s_0)), the number of steps made by motor-A prior to recognition is a random variable (y_1) dependent on the number of steps made by motor-B - (x^*)

$$p(x^*, y_1|s_{switch}; \theta) = p(y_1|x^*, s_{switch}; \theta)p(x^*, s_{switch}; \theta)$$

Then

$$p(x^*, y_1, y_2|s_{term}; \theta) = p(y_2|s_{term}; \theta)p(y_1|x^*, s_{switch}; \theta)p(x^*, s_{switch}; \theta) \tag{3.5}$$

Now that Eq. 3.5 has been partitioned into three independent components, it is possible to derive those separately and then combine them under Eq. 3.5 and then calculate $p(x^*, y^*, s_{term})$ using Eq. 3.4

$$p(x^*, y^*|s_{term}) = \sum_{k=1}^{y^*} p(y^* - k|s_{term}; \theta)p(k|x^*, s_{switch}; \theta)p(x^*, s_{switch}; \theta) \tag{3.6}$$

3.3.1 x-component

First, let us derive the probability distribution of the number of transitions x of motor-B before it recognises a SWITCH ($x = x^*$). By the time motor-B arrives at x^*, motor-A has made y_1 steps and the external state of SM is now

$$(x^*, y_1), x^* \in X^* = \{x_1^*, ..., x_I^*\}, y_1 \in \mathcal{X}$$

where X^* is a set of position of SWITCHes.

Let $p(x = x_i^*, s_{switch}; \theta)$ denote the probability of SM recognising a SWITCH. As a result of recognition, motor-B would stall. According to the transition rules formulated above, this is a geometric distribution with a probability of recognition of an individual SWITCH p_χ assuming all SWITCH sites are equally likely to be detected and totally independent. Then the probability of recognising the i^{th} SWITCH from the point of loading of the SM is

$$p(x_i^*, s_{switch}; \theta) = p_\chi q_\chi^{i-1} = \mathcal{G}(p_\chi, i) = \mathcal{G}(1 - q_\chi, i) \tag{3.7}$$

3.3.2 The y-component dependent on x: y_1

The second component of Eq. 3.5, $p(y_1|x, s_{switch}; \theta)$, describes how far motor-A (y) is likely to advance along the sequence, given SM has recognised a SWITCH at x_i^*. Let us denote the probability of "success" in each trial (iteration of MC) which is one of the two possible outcomes whereby motor-B skips a step. This occurs with probability p_-. By the time x_i^* "failures" occur, the total number of trials y_1 is a random variable that follows a negative binomial distribution with parameter p_-.

$$p(y_1|x_i^*, s_{switch}; p_-) = \binom{y_1}{y_1 - x_i^*} p_-^{x_i^*} p_+^{y_1 - x_i^*} := \mathcal{B}^-(p_-, y_1, x_i^*) \tag{3.8}$$

In fact, p_- is a good approximation of the average speed ratio of motors A and B ($p_- = V_B/V_A$). If the speed of motor-A was equal to that of motor-B then $p_- = 1$ and $p_+ = 0$ and $y_1 = x^*$.

3.3.3 The second y-component independent of x: y_2

From the structure of MC (Fig. 3.2), the second component y_2 follows a geometric distribution with parameter p_s

$$p(y_2|s_{term};\theta) = p_s q_s^{y_2} = \mathcal{G}(p_s, y_2) \tag{3.9}$$

Then, by combining Eq. 3.8 and Eq. 3.9 we derive the distribution for the total number of steps made by motor-B until dissociation $y^* = y_1 + y_2$ given x_i^*.

$$p(y^*|x_i^*;\theta) = \sum_{y_1=1}^{y} p(y^*-y_1, s_{term};\theta)p(y_1|x_i^*, s_{switch};\theta) = \sum_{y_1=1}^{y^*} \mathcal{B}^-(p_-, y_1, x_i^*)\mathcal{G}(p_s, y^*-y_1) \tag{3.10}$$

3.3.4 Assembling the probability distribution over absorbing states of MC

Plugging Eq. 3.10 and Eq. 3.7 into Eq. 3.5, we obtain

$$p(x_i^*, y^*|s_{term};\theta) = \mathcal{G}(p_\chi, i)\sum_{k=1}^{y^*} \mathcal{B}^-(p_-, k, x_i^*)\mathcal{G}(p_s, y^* - k) \tag{3.11}$$

where $x_i^* \in X^*$ and $y^* \in \mathcal{X}$.

3.4 Some properties affecting the distribution of the output (x^*, y^*) of SM

Definition 11 (Segment Output (OS))**.** *The sequence segment output (OS) is the interval of the sequence between x and y steps from the position of loading, produced by the time MC reaches its absorbing state (x^*, y^*, s_{term}).*

$$\boldsymbol{l} = (x^*, y^*)$$

The length of OS is

$$L = |\boldsymbol{l}| = y^* - x^* + 1$$

3.4.1 Mean length of the segment

Theorem 6 (Mean length of the sequence segment)**.**

$$\mathrm{E}(L) = \mathrm{E}(y^* - x^* + 1) = \mathrm{E}(\tau_1) + \mathrm{E}(\tau_2) \approx p_+/p_- \langle x^* \rangle + 1/p_s$$

where $\langle x^* \rangle$ is geometric mean of the positions of the SWITCHes

$$\langle x^* \rangle \coloneqq \frac{p_\chi}{1 - q_\chi^I} \sum_{i=1}^{I} x_i^* q_\chi^{i-1} \tag{3.12}$$

Proof. Denote $\tau_1 = y_1 - x + 1$, $\tau_2 := y_2$ and $L = \tau_1 + \tau_2$
From Eq. 3.9, it follows that the expectation of τ_2 is

$$\mathrm{E}(\tau_2) \approx \sum_{\tau_2=0}^{\infty} \tau_2 \mathcal{G}(p_s, \tau_2) = 1/p_s \tag{3.13}$$

Using Eq. 3.8, let us find the expectation of τ_1 conditional on x_i^*

$$\mathrm{E}(\tau_1|x^*) \approx \sum_{\tau_1=0}^{\infty} \tau_1 p(\tau_1|x^*) = \sum_{\tau_1=0}^{\infty} \tau_1 \mathcal{B}^-(p_-, \tau_1, x^*) = p_+/p_- x^*$$

The unconditional expectation of τ_1 can then be calculated by taking the average over all the positions of SWITCH motifs $x_1, ..., x_I$

$$\mathrm{E}(\tau_1) = \mathrm{E}_{x^*}(\mathrm{E}(\tau_1|x^*)) = \mathrm{E}_{x^*}(p_+/p_- x^*) = p_+/p_- \mathrm{E}(x^*) \approx p_+/p_- \frac{\sum_{i=1}^{I} p(x_i^*) x_i^*}{\sum_{i=1}^{I} p(x_i^*)}$$

$$= p_+/p_- \frac{\sum_{i=1}^{I} x_i^* q_\chi^{i-1}}{\sum_{i=1}^{I} q_\chi^{i-1}} = p_\chi p_+/p_- \frac{\sum_{i=1}^{I} x_i^* q_\chi^{i-1}}{1 - q_\chi^I}$$

since $p(x_i^*)$ follows a geometric distribution with parameter p_χ (Eq. 3.7).
This result can be rewritten as:

$$\mathrm{E}(\tau_1) \approx p_+/p_- \langle x^* \rangle \tag{3.14}$$

Eventually, substituting Eq. 3.13 and Eq. 3.14 into $\mathrm{E}(L) = \mathrm{E}(\tau_1) + \mathrm{E}(\tau_2)$ (remembering that τ_1 and τ_2 are independent)

$$\mathrm{E}(L) = \mathrm{E}(\tau_1) + \mathrm{E}(\tau_2) \approx p_+/p_- \langle x^* \rangle + 1/p_s$$

Note that this result is the asymptotic expectation of L in the limit of an infinitely large sequence span $X \to \infty$. □

3.4.2 Density of sequence SWITCHes

Adding additional SWITCHes leftmost

It is of interest to investigate how an increased density of SWITCH-sites affects the expected length $\mathrm{E}(L)$ of the segment output of the SM. First, let us consider adding an additional SWITCH leftmost, which will increase the overall density of SWITCH-sites. Earlier SWITCH sites have higher (because they have low rank) but shorter (because their coordinate is smaller) plateaux. A qualitative consequence is described in the following lemma.

Lemma 1 (Insertion of an additional SWITCH non-rightmost (Vincent Danos))**.** *Given the sequence of SWITCHes* $X^* = \{x_1^*, ..., x_k^*, ..., x_I^*\}$ *inserting an additional SWITCH (non-rightmost) decreases* $\mathrm{E}(L)$*; so does moving a SWITCH left (i.e. closer to the initial position of SM, for example when one of the SWITCHes was moved left by* λ*,* $\bar{x}_k^* = x_k^* - \lambda$*, so that the new sequence of SWITCHes is* $\bar{X}^* = \{x_1^*, ..., \bar{x}_k^*, ..., x_I^*\}$ *)*

Proof.

$$\mathrm{E}(L^*) - \mathrm{E}(L) = p_+/p_-(\langle \bar{x}^* \rangle - \langle x^* \rangle) = \frac{p_\chi p_+/p_-}{1 - q_\chi^I}\Big(\sum_{x^* \in \bar{X}^*} x_i^* q_\chi^{i-1} - \sum_{x^* \in X^*} x_i^* q_\chi^{i-1}\Big) =$$

$$= \frac{p_\chi p_+/p_-}{1 - q_\chi^I}(\bar{x}_k^* - x_k^*) = -\frac{p_\chi p_+/p_-}{1 - q_\chi^I}\lambda < 0$$

□

Switch-motif clusters

Now, let us consider the cluster of SWITCHes, which by the way also contributes to the increased overall density of SWITCHes on the sequence

Theorem 7 (Insertion of a SWITCH-cluster)**.** *Let two sets of SWITCHes be X^* and X_1^*, $X^* = \{x_1^*, x_2^*, ..., x_I^*\}$ and $\bar{X}^* = \{x_1^*, x_1^*, ..., x_1^*, x_2^*, ..., x_I^*\}$, where the first SWITCH x_1^* is represented k times.*
Then (1)

$$\mathrm{E}(\bar{\tau}_1) - \mathrm{E}(\tau_1) \approx p_\chi p_+/p_-(1-q_\chi^{k-1})\Big(q_\chi x_1^* - \langle x_{2...I}^*\rangle\Big)$$

when $I \to \infty$
(2) For periodic SWITCHes with period x_1^*: $x_i^* = ix_1^*$, $i = 1, ..., I$

$$\Delta E(\tau_1) = \mathrm{E}(\bar{\tau}_1) - \mathrm{E}(\tau_1) \approx -p_+/p_- q_\chi/p_\chi(1-q_\chi^{k-1})x_1^*$$

(3) and the relative difference is

$$\epsilon\mathrm{E}(\tau_1) = -(1-q_\chi^{k-1})q_\chi$$

Proof. The geometric mean of the SWITCH locations is described by (Eq. 3.12):

$$\langle x^*\rangle = \frac{p_\chi}{1-q_\chi^I}\sum_{i=1}^{I} x_i^* q_\chi^{i-1}$$

Here, I only consider the case when I is very large.
Since $\lim_{I\to\infty}\frac{1}{1-q_\chi^I} = 1$

$$\mathrm{E}(\tau_1) = p_\chi p_+/p_-\langle x_i^*\rangle = p_+/p_-\frac{p_\chi}{1-q_\chi^I}\sum_{i=1}^{I} x_i^* q_\chi^{i-1} \approx p_\chi p_+/p_-\sum_{i=1}^{I} x_i^* q_\chi^{i-1}$$

This is a good approximation when $q_\chi \leq 0.8$ and $I \geq 10$ (Fig. 3.3).

Then,

$$\mathrm{E}(\tau_1^*) - \mathrm{E}(\tau_1) \approx p_\chi p_+/p_- \sum_{i=2}^{I} x_i^* \Big(q_\chi^{i+k-2} - q_\chi^{i-1} \Big) + p_\chi p_+/p_- x_1^* \sum_{i=1}^{k} \Big(q_\chi^{i-1} - 1 \Big) =$$

$$= p_\chi p_+/p_- \Big[\sum_{i=2}^{I} x_i^* \Big(q_\chi^{i+k-2} - q_\chi^{i-1} \Big) + x_1^* \sum_{i=2}^{k} q_\chi^{i-1} \Big] =$$

$$= p_\chi p_+/p_- \Big[(q_\chi^{k-1} - 1) \sum_{i=2}^{I} q_\chi^{i-1} x_i^* + x_1^* \sum_{i=1}^{k-1} q_\chi^i \Big] =$$

$$= p_\chi p_+/p_- \Big[(q_\chi^{k-1} - 1) \sum_{i=1}^{I-1} q_\chi^i x_{i+1}^* + x_1^* q_\chi \frac{1 - q_\chi^{k-1}}{1 - q_\chi} \Big] =$$

$$= p_\chi p_+/p_- (1 - q_\chi^{k-1}) \Big[\frac{q_\chi x_1^*}{p_\chi} - \sum_{i=1}^{I-1} q_\chi^i x_{i+1}^* \Big]$$

Remembering the definition of a geometric mean of SWITCH locations (Eq. 3.12):

$$p_\chi \sum_{i=1}^{I-1} q_\chi^i x_{i+1}^* = p_\chi \sum_{i=2}^{I} q_\chi^{i-1} x_i^* \coloneqq \langle x_{2\ldots I}^* \rangle$$

Now, let us replace the sum by this new notation

$$\mathrm{E}(\tau_1^*) - \mathrm{E}(\tau_1) \approx p_+/p_- (1 - q_\chi^{k-1}) \Big(q_\chi x_1^* - \langle x_{2\ldots I}^* \rangle \Big) \tag{3.15}$$

Let the sequence of SWITCHes be periodic with period x_1^*. Then

$$\langle x_{2\ldots I}^* \rangle = p_\chi x_1^* \sum_{i=2}^{I} q_\chi^{i-1} i \approx x_1^* (2q_\chi - q_\chi^2/p_\chi) \; when \; I \to \infty$$

Then substituting this into Eq. 3.15 we obtain

$$\mathrm{E}(\bar{\tau}_1) - \mathrm{E}(\tau_1) \approx p_+/p_- (1 - q_\chi^{k-1})(q_\chi x_1^* - x_1^*(2q_\chi - q_\chi^2/p_\chi)) = -p_+/p_- (1 - q_\chi^{k-1}) x_1^* (q_\chi + q_\chi^2/p_\chi)$$

$$\mathrm{E}(\tau_1^*) - \mathrm{E}(\tau_1) \approx -p_+/p_-(1-q_\chi^{k-1})x_1^* q_\chi/p_\chi$$

For the periodically located x^*s expected τ_1 when $I \to \infty$

$$\mathrm{E}(\tau_1) \approx p_+/p_- x_1^*/p_\chi$$

Then the relative difference is

$$\epsilon\mathrm{E}(\tau_1) = \frac{\Delta\mathrm{E}(\tau_1)}{\mathrm{E}(\tau_1)} \approx \frac{-p_+/p_-(1-q_\chi^{k-1})x_1^* q_\chi/p_\chi}{p_+/p_- x^*/p_\chi} = -(1-q_\chi^{k-1})q_\chi \tag{3.16}$$

□

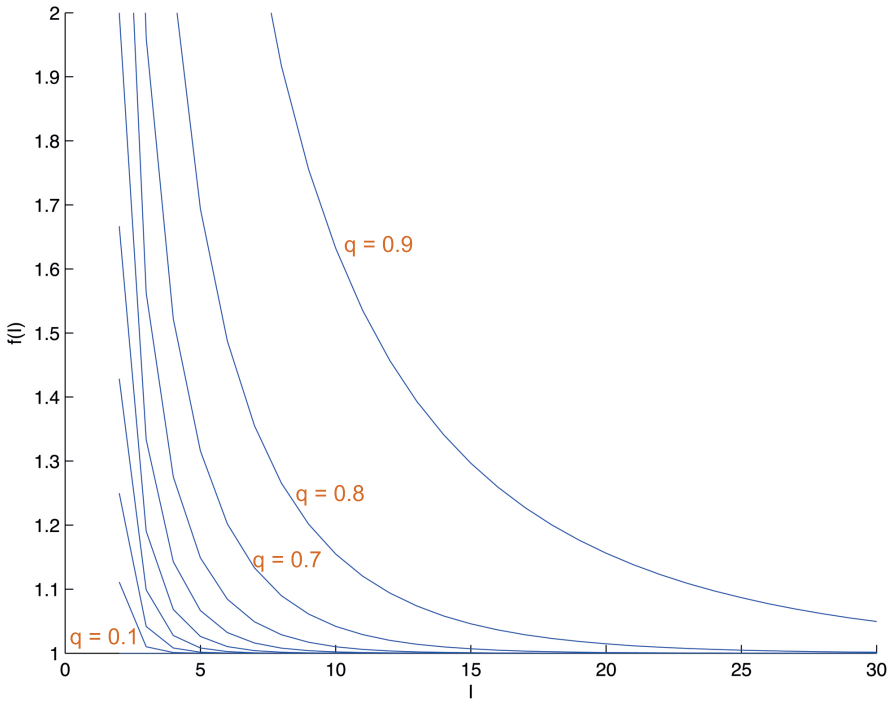


Figure 3.3. $f(I) = \frac{1}{1-q_\chi^{I-1}}$ plotted for a range of parameters $q_\chi = [0.1 : 0.1 : 0.9]$. $f(I)$ approaches one in the limit of large I $f(I) \xrightarrow{I\to\infty} 1$. The smaller q_χ (and larger p_χ), the faster f converges to one.

Therefore, it follows from Theorem 7 that the difference in the mean length of OS saturates at a constant, so inserting additional SWITCHes in the array has less and less impact on the estimated mean OS length and its position accordingly (Fig. 3.4). This difference quickly saturates at just a few repeats for small q_χ. Even for $p_\chi = 0.5$ the saturation already occurs at 4-5 repeats.

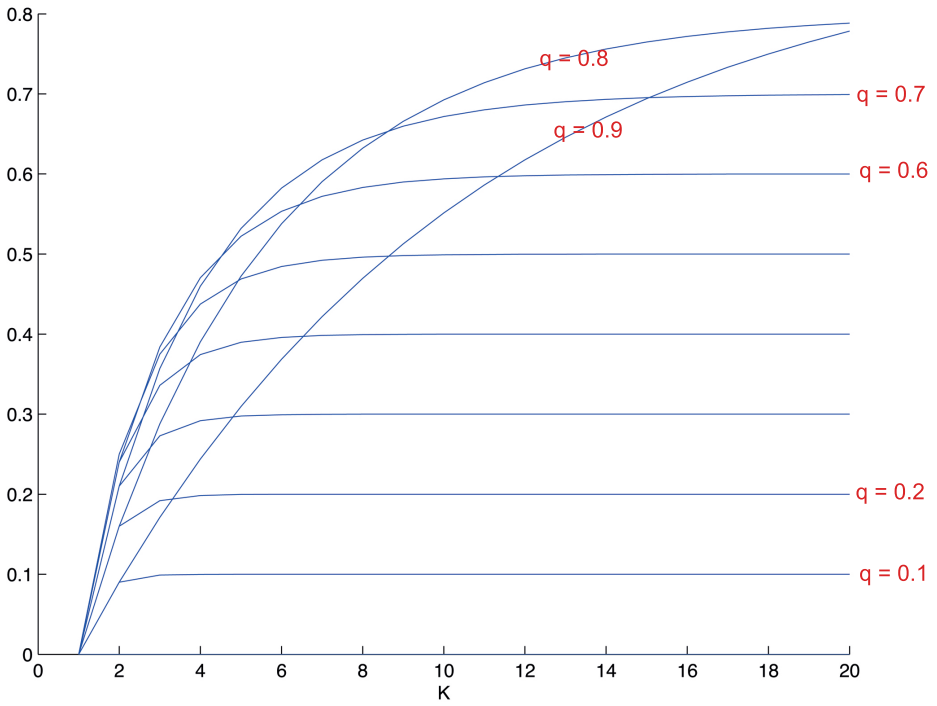


Figure 3.4. The relative difference in the mean length of the output with respect to the mean of the first segment of the output τ_1 after having inserted a SWITCH array with k SWITCHes (Eq. 3.16) plotted for a range of parameters $q_\chi = [0 : 0.1 : 1]$.

3.4.3 Truncation of the MC state space

The number of states of the SM is limited to $x \in \mathcal{X}$, $y \in \mathcal{X}$ because of truncation, therefore it is essential to estimate the error introduced by the truncation of the state space.

Runaways

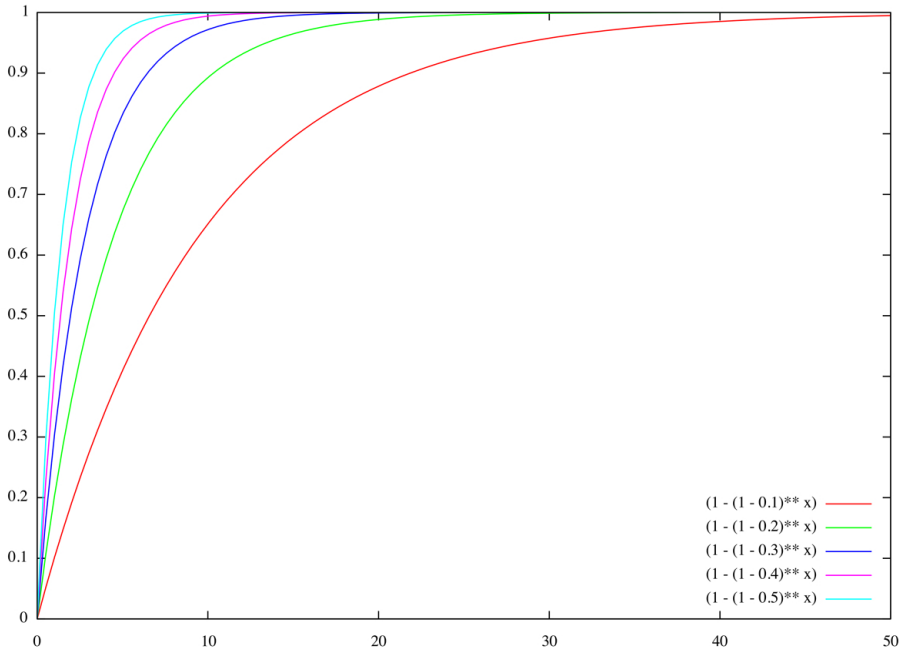


Figure 3.5. Probability of success (falloffs) in finite time as a function of the number of trials (number of SWITCHes) for various values of p_χ.

When I is finite there is a nonzero probability that SM exits without recognising any SWITCH within the truncated segment of the sequence. Therefore there is always an error associated with truncation of the state space $\mathcal{X}$ which would embrace only a finite number of SWITCH sites I.

$$p(Recognised\ SWITCH\ is\ within\ I\ SWITCHes\ from\ the\ start) = p_\chi \sum_{n=1}^{I} q_\chi^{n-1} = 1 - q_\chi^I$$

$$p(\text{“}runaway\text{''}) = 1 - (1 - q_\chi^I) = q_\chi^I$$

The probability of runaway goes to zero as the number of SWITCHes within the truncation goes to infinity (Fig. 3.5).

$$\lim_{I \to \infty} p(\text{“}runaway\text{''}) = 0$$

Lemma 2. *Random variable ξ is distributed geometrically with parameter p on $\mathbb{N}^+ = [1, \infty)$*

$\xi \sim \mathcal{G}(p)$

Let $\mathcal{X} = (1, ..., X)$ be a truncation of $\mathbb{N}^+$: $\mathcal{X} \subset \mathbb{N}^+$

Then

$$\Delta \mathrm{E}(\xi) = \mathrm{E}_{\mathbb{N}^+}(\xi) - \mathrm{E}_{\mathcal{X}}(\xi) = \frac{X q^X}{1 - q^X}$$

Proof.

$$\mathrm{E}_{\mathbb{N}^+}(\xi) = p \sum_{i=1}^{\infty} q^{i-1} i = p \frac{d}{dq} \Big[\sum_{i=0}^{\infty} q^i \Big] = p \frac{d}{dq} 1/(1-q) = 1/p$$

$$\mathrm{E}_{\mathcal{X}}(\xi) = \frac{\sum_{i=1}^{X} q^{i-1} i}{\sum_{i=1}^{X} q^{i-1}} = p/(1-q^X) \frac{d}{dq} \Big[\sum_{i=1}^{X} q^i \Big] =$$

$$p/(1-q^X) \frac{d}{dq} \frac{q(1-q^X)}{1-q} = 1/p - \frac{X q^{X-1}}{1-q^X}$$

$$\Delta \mathrm{E}(\xi) = \mathrm{E}_{\mathbb{N}^+}(\xi) - \mathrm{E}_{\mathcal{X}}(\xi) = \frac{X q^X}{1 - q^X}$$

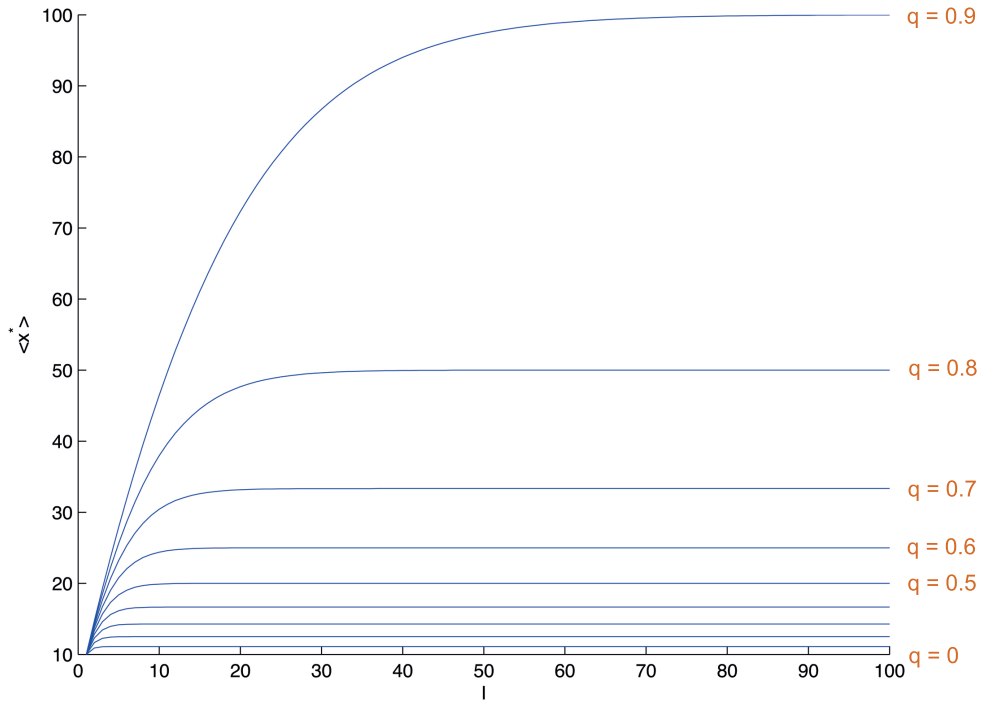


Figure 3.6. The periodic sequence of SWITCHes $X^* = [10, 20, 30, ...]$, geometric mean of their positions as a function of the total number of SWITCHes included in truncation $\langle x^* \rangle (I) = p_\chi / (1 - q_\chi^I) \sum_{i=1}^{I} x_i^* q_\chi^{i-1}$ for a range of probabilities of SWITCH recognition $q_\chi = [0 : 0.1 : 1]$.

□

Theorem 8 (Error associated with the truncation of SWITCH sequence)**.** *Suppose* $x^* = i x_1^*;\ i \in \mathbb{N}^+$

Consider the truncation of this sequence at I*:* $i = 1, ..., I$.

Then

$$\epsilon E(\tau_1)|_{[I,\infty)} = p_\chi \frac{I q_\chi^I}{1 - q_\chi^I}$$

Proof. Using the result obtained in Lemma 2

$$\Delta\langle x^*\rangle|_{[I,\infty)} = \langle x^*\rangle|_\infty - \langle x^*\rangle|_I = p_\chi \sum_{i=1}^{\infty} i x_1^* q_\chi^{i-1} - p_\chi/(1-q_\chi^I) \sum_{i=1}^{I} i x_1^* q_\chi^{i-1} = x_1^* \frac{I q_\chi^I}{1-q_\chi^I}$$

Therefore

$$\epsilon E(\langle x^*\rangle|_{[I,\infty)}) = \frac{\Delta\langle x^*\rangle|_{[I,\infty)}}{\langle x^*\rangle|_\infty} = p_\chi \frac{I q_\chi^I}{1-q_\chi^I}$$

and consequently

$$\epsilon E(\tau_1)|_{[I,\infty)} = p_\chi \frac{I q_\chi^I}{1-q_\chi^I}$$

□

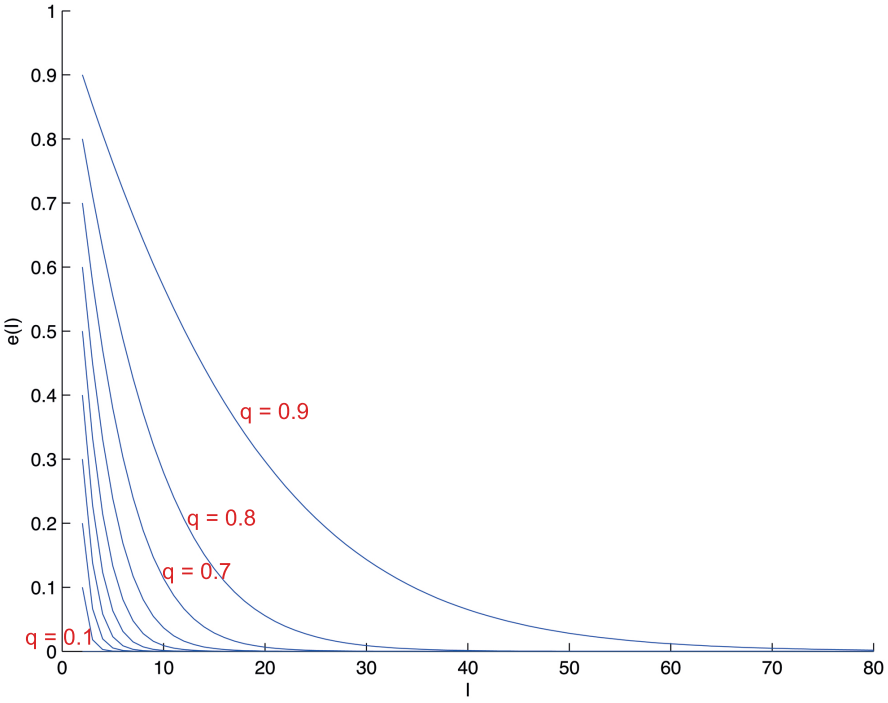


Figure 3.7. The relative error in the estimation of $E(\tau_1)$ as a function of the total number of SWITCHes included in truncation $\epsilon E(\tau_1) = p_\chi \frac{I q_\chi^I}{1-q_\chi^I}$ for a range of probabilities of SWITCH recognition $q_\chi = [0 : 0.1 : 1]$.

According to Theorem 8 the error decreases as we include more SWITCHes, and the rate of error drop increases with the higher probability of SWITCH recognition (p_χ) (Fig. 3.7). So in order to minimise this error we need to be careful about how many SWITCH sites to include, ideally as many SWITCHes as possible, especially when p_χ is expected to be in the low range.

Fixed number of SWITCHes, truncation of the sequence span X

Consider the truncation that spans only one SWITCH ($I = 1$). Also, the distance from the initial position x_0 to the truncation point is X. And there are no other SWITCHes in between the truncation point and x_1^*:

$$x_2^* \geq X$$

Also, let us require that the truncation point goes beyond the average extent of the first segment formed before recognition of a SWITCH occurs (τ_1)

$$X > \mathrm{E}(\tau_1) + x_1^* = p_+/p_- x_1^* + x^* = x_1^*/p_-$$

Lemma 3 (The error associated with the truncation of the sequence span X (Single SWITCH))**.** *The error on the position of the second part of the OS (τ_2) is*

$$\epsilon \mathrm{E}(\tau_2)|_{[X,\infty)} = p_s \frac{(X - x_1^*/p_-) q_s^{X - x_1^*/p_-}}{1 - q_s^{(X - x_1^*/p_-)}}$$

Proof. Conditional expectation of τ_2

$$\mathrm{E}(\tau_2|\tau_1) = p_s \sum_{i=1}^{X-\tau_1} q_s^{i-1} i$$

Expectation of τ_2

$$\mathrm{E}(\tau_2) = \mathrm{E}_{\tau_1}\mathrm{E}(\tau_2|\tau_1) = p_s \sum_{i=1}^{X-x_1^*/p_-} q_s^{i-1} i$$

Using the result of Lemma 2

$$\Delta\mathrm{E}(\tau_2)|_{[X,\infty)} = \frac{(X - x_1^*/p_-)q_s^{X-x_1^*/p_-}}{1 - q_s^{(X-x_1^*/p_-)}}$$

and the error associated with truncation is

$$\epsilon\mathrm{E}(\tau_2)|_{[X,\infty)} = p_s\frac{(X - x_1^*/p_-)q_s^{X-x_1^*/p_-}}{1 - q_s^{(X-x_1^*/p_-)}}$$

□

Now, let us consider several SWITCHes ($\{x_1^*, x_2^*, ..., x_I^*\}$) within the span and as previously demand that

$$X > \mathrm{E}(\tau_1|x_I^*) + x_I^* = x_I^*/p_-$$

$$x_I^*/p_- < X < x_{I+1}^*$$

Theorem 9 (The error associated with the truncation of the sequence span X)**.**

$$\epsilon\mathrm{E}(\tau_2)|_{[X,\infty)} < p_s\frac{(X - x_I^*/p_-)q_s^{(X-x_I^*/p_-)}}{1 - q_s^{X-x_I^*/p_-}}$$

Proof. Given the result of Lemma 3, the error conditional on x^* being recognised is

$$\Delta E(\tau_2|x^*)|_{[X,\infty)} = \frac{(X - x^*/p_-)q_s^{(X-x^*/p_-)}}{1 - q_s^{(X-x^*/p_-)}}$$

The cumulative error is

$$\mathrm{E}_{x^*}(\Delta E(\tau_2|x^*)|_{[X,\infty)}) = \mathrm{E}_{\tau_1}\Big(\frac{(X - x_i^*/p_-)q_s^{X-x_i^*/p_-}}{1-q_s^{X-x_i^*/p_-}}\Big) = p_\chi/(1-q_\chi^I)\sum_{i=1}^{I} q_\chi^{i-1}\frac{(X - x_i^*/p_-)q_s^{X-x_i^*/p_-}}{1-q_s^{X-x_i^*/p_-}}$$

$$< p_\chi/(1-q_\chi^I)\sum_{i=1}^{I} q_\chi^{i-1}\frac{(X - x_I^*/p_-)q_s^{X-x_I^*/p_-}}{1-q_s^{X-x_I^*/p_-}} = p_\chi/(1-q_\chi^I)\frac{(X - x_I^*/p_-)q_s^{X-x_I^*/p_-}}{1-q_s^{X-x_I^*/p_-}}(1-q_\chi^I)/p_\chi$$

$$\Delta\mathrm{E}(\tau_2) < \frac{(X - x_I^*/p_-)q_s^{X-x_I^*/p_-}}{1-q_s^{X-x_I^*/p_-}}$$

$$\epsilon\mathrm{E}(\tau_2)|_{[X,\infty)} < p_s\frac{(X - x_I^*/p_-)q_s^{X-x_I^*/p_-}}{1-q_s^{X-x_I^*/p_-}}$$

□

The consequence of Theorem 9 is that

$$\lim_{X\to\infty} \epsilon\mathrm{E}(\tau_2)|_{[X,\infty)} = 0$$

Hence the error associated with the estimation of τ_2 can be reduced by choosing the truncation point X as far as possible from the last captured SWITCH (I^{th} SWITCH), ideally strictly at the following SWITCH and choose I where the distance between x_{I+1}^* and x_I^* is large.
For example, for $q_s = 0.01$ and the distance to the next SWITCH should be

$$x_{I+1}^* - x_I^*/p_- > 500$$

to reduce the relative error to 5% (Fig. 3.8)

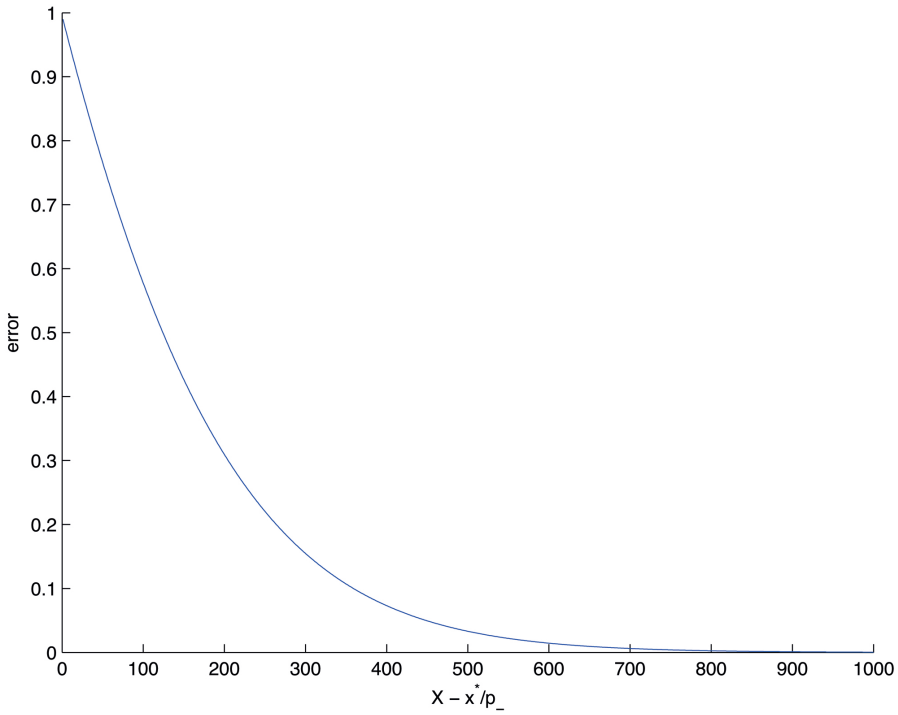


Figure 3.8. The relative error in the estimation of $E(\tau_2)$ as a function of the distance between the last captured SWITCH x_I^* and the truncation point X (Eq. 9) for $p_s = 0.01$.

3.4.4 Variance of L

This paragraph features the variance of the mass of the sequence segment $\mathrm{Var}(L)$. Because the two random processes of accumulation of mass before (τ_1) and after SWITCH recognition (τ_2) are independent, it follows that

$$\mathrm{Var}(L) = \mathrm{Var}(\tau_1) + \mathrm{Var}(\tau_2)$$

$$\mathrm{Var}(\tau_2) = q_s/p_s^2$$

$$\mathrm{Var}(\tau_1) = \mathrm{E}(\tau_1^2) - \mathrm{E}(\tau_1)^2$$

According to the previously derived relation for $\mathrm{E}(\tau_1)$

$$\mathrm{E}(\tau_1^2) = \mathrm{E}(\mathrm{E}(\tau_1^2|x_i^*)) = (p_+/p_-)^2 \frac{\sum_{i=1}^{I} x_i^{*2} p_\chi q_\chi^{i-1}}{1 - q_\chi^I} = (p_+/p_-)^2 \langle x^{*2} \rangle$$

where

$$\langle x^{*2} \rangle \coloneqq \frac{p_\chi}{1 - q_\chi^I} \sum_{i=1}^{I} x_i^{*2} q_\chi^{i-1}$$

$$\mathrm{E}(\tau_1) = p_+/p_- \langle x^* \rangle \;\&$$

$$\mathrm{E}(\tau_1)^2 = (p_+/p_-)^2 (\langle x^* \rangle)^2 \Rightarrow$$

$$\mathrm{Var}(\tau_1) = \mathrm{E}(\tau_1^2) - \mathrm{E}(\tau_1)^2 = (p_+/p_-)^2 \langle x^{*2} \rangle - (p_+/p_-)^2 \langle x^* \rangle^2 =$$

$$= (p_+/p_-)^2 (\langle x^{*2} \rangle - (\langle x^* \rangle)^2)$$

Thus, using the new notation

$$\mathrm{Var}(L) = (p_+/p_-)^2 (\langle x^{*2} \rangle - (\langle x^* \rangle)^2) + q_s/p_s^2$$

Chapter 4

MC model fit to ideal pileup sequencing data

In this chapter I attempt to simulate an idealised version of pileup sequencing data (synthetic data), devoid of certain constraints of a real experiment, using some parametrically defined Markov chain that generates a range of output sequence segments, whose distribution is $p(x^*, y^*; \theta)$ (Chapter 3). Also, I will employ statistical methods to solve a reverse problem: derive the parameters of MC from the simulated data.

4.1 Population scale output of SM

To mimic the real experiment of the sequencing data acquisition where the sequence outputs are generated by a large number of independent molecules, let us consider a population of identical SMs defined in Chapter 3 and a snapshot of their evolution. Each of them is captured by the snapshot either in the transient

or absorbing state with a certain sequence output accumulated by the time of the snapshot, which is a random output of the Markov chain of SM and can be mathematically represented as a random vector $\boldsymbol{l} = (x, y)$, where x is a start position and y is an end position of the sequence segment output.

Definition 12 (Output segment (OS))**.** *The output sequence segment of SM is a random vector, where x is a start position and y is an end position of the OS*

$$\boldsymbol{l} = (x, y)$$

Definition 13 (Map of the output segment (MOS))**.** *The map of the output OS to the reference sequence can be mathematically expressed as*

$$\mathbf{1}_{xy} \coloneqq (00...01_x...1_y0...00)$$

$$\forall i \in [x, y]\ \mathbf{1}_{xy}(i) = 1,\ \forall i \notin [x, y]\ \mathbf{1}_{xy}(i) = 0$$

The result of mapping multiple OS generated by multiple unsynchronised SMs alike can be mathematically viewed as a sum of multiple (N) OS vectors.

Definition 14 (Ideal data (ID))**.** *The cumulative output of a population of SMs that we would be able to read in ideal conditions, without sampling and fragmentation (hereafter referred to as "ideal data") is*

$$\boldsymbol{D} = \sum_{i=1}^{N} \mathbf{1}_{xy}$$

where N is the number of OSes mapped. Note that in the limit of an infinitely large number of OSes, $\boldsymbol{D}/N$ is exactly the expectation of MOS

$$\mathrm{E}(\mathbf{1}_{xy}) = \lim_{N\to\infty} 1/N \sum_{i=1}^{N} \mathbf{1}_{xy}$$

Since the absorbing state of the SM is the longest lived, the measurement of the OS in the absorbing state of SM should make the largest contribution to the ID and the transient states would be present at a negligible level.

$$\boldsymbol{D} = \sum_{i=1}^{N} \mathbf{1}_{xy} \approx \sum_{i=1}^{N^*} \mathbf{1}_{x^*y^*}$$

where (x^*, y^*) are OSes generated by the SMs in the terminal state (in the absorbing state of MC).

4.2 Assumptions

Here I will introduce the assumptions necessary to build a model of a ChIP-Seq experiment, which captures essential features of data acquisition with a few limitations later clarified in Chapter 5.

First, when the Markov chain of SM arrives in its absorbing state, it outputs an OS of some type (x^*, y^*), $x^* \in \mathcal{X}, y^* \in \mathcal{X}$ (Chapter 3). A population of SMs would output OSes of multiple types. Each OS in the population output is in turn fragmented uniformly without bias into fixed size fragments of length w. The fragments collected from a whole population of OSes will be further referred to as "pool".

A few fragments are then isolated during the course of the experiment (later referred to as "sample"). An assumption is being made that the total number of fragments N_s randomly sampled from the pool is large ($N_s \to \infty$), yet the sample size is very small compared to the pool size ($N_s \ll N_{pool}$,), and hence the fragments can be seen as drawn independently and with replacement. These assumptions will be discussed in Chapter 5.

Definition 15 (Fragment of type (x, y))**.** *A fragment is of type (x, y) if it was generated by fragmentation of an OS of type (x, y).*

$$(x, y)_f = \{(x', y'); (x', y') \subset (x, y) \ \& \ y' - x' + 1 = w\}$$

w is the fixed size of a fragment.

The expected fraction of (x, y)-OS in the pool before fragmentation equals the probability of SM's MC arriving at an (x, y)-absorbing state $S = (x, y, s_{term})$

$$\mathrm{E}\Big(\frac{N_{xy}^{pool}}{N^{pool}}\Big) = p(x, y, s_{term}) \tag{4.1}$$

I omit the third component and from now on write $p(x, y)$ instead of $p(x, y, s_{term})$.

Accordingly, the fraction of (x, y)-fragments in the pool after fragmentation is

$$\mathrm{E}\Big(\frac{N_{f-xy}^{pool}}{N_f^{pool}}\Big) = \frac{p(x, y)[(y - x + 1)/w]}{\sum_{xy} p(x, y)[(y - x + 1)/w]} \tag{4.2}$$

since a single (x, y)-OS donates on average $[(y - x + 1)/w$ fragments to the pool.

When a random fragment is drawn from the pool, the probability that it is of (x, y)-type is equal to its frequency of occurrence in the pool:

$$p((x, y)_f) = \frac{p(x, y)[y - x + 1/w]}{\sum_{x,y} p(x, y)[y - x + 1/w]} \tag{4.3}$$

Because of the assumption of considerable dilution of the random sample, I model the fragment draws as totally independent from each other, which means that the event of drawing one fragment from the pool does not affect the composition of the pool and hence the probability of drawing another fragment of any type. In this case the draws can be viewed as made with replacement.

Given the assumptions made above I utilise a multinomial distribution as a model for the distribution of the number of fragments across their types (x, y) in the sample of N_s independently drawn fragments from the pool.

$$\boldsymbol{n}_{\mathcal{X}\times\mathcal{X}} \sim MN(N_s, \boldsymbol{p}_{\mathcal{X}\times\mathcal{X}}) \tag{4.4}$$

where $\boldsymbol{p}_{\mathcal{X}\times\mathcal{X}} = (p((1,1)_f), p((1,2)_f), ..., p((X,X)_f))$
and $\boldsymbol{n}_{\mathcal{X}\times\mathcal{X}} = (n_{(1,1)}, n_{(1,2)}, ..., n_{(X,X)})$;
Then, the expected number of fragments of each type present in the sample is

$$\mathrm{E}(\boldsymbol{n}_{\mathcal{X}\times\mathcal{X}}) = N_s\boldsymbol{p}_{\mathcal{X}\times\mathcal{X}} \tag{4.5}$$

When the sample N_s goes to infinity, the number of fragments of each type converges to their mean $\mathrm{E}(n_{(xy)})$

$$\boldsymbol{n}_{\mathcal{X}\times\mathcal{X}} \to \mathrm{E}\boldsymbol{n}_{\mathcal{X}\times\mathcal{X}} = N_s\boldsymbol{p}_{\mathcal{X}\times\mathcal{X}} \; when \; N_s \to \infty \tag{4.6}$$

According to the protocol of acquisition of pileup sequencing data, each fragment that belongs to the sample donates a single position chosen uniformly at random, which is subsequently turned into a single hit in the count data. For some fragment (x', y')

$$\xi = \mathcal{U}_{[x',y']} \tag{4.7}$$

Upon a successful mapping of this (x', y')-fragment the number of hits at position ξ increases by one

$$n_\xi \coloneqq n_\xi + 1 \tag{4.8}$$

Definition 16 (Ideal Pileup Data (IPD))**.** *Multiple mapping events aggregated together, such as described above, of a sample of N_s fragments constitute "pileup data"*

$$\boldsymbol{n} = (n_1, n_2, ..., n_X)$$

4.3 Simulation of IPD

4.3.1 Algorithm

In this subsection I simulate IPD by incorporating the assumptions outlined in the previous section. The setup of the simulation combines

- The sample size (total number of fragments in the sample) N_s.
- The fragment size w.
- The values of the parameters used to construct the transition probabilities of MC θ_{synth}.
- The probability of generating an OS of type (x, y) in the absorbing state of MC being a function of this particular set of parameters - $p(x, y; \theta)$. This probability will be used as the model for the fraction of (x, y)-OS in the initial pool of OSes.
- Truncation of the state space at X, $\mathcal{X} = (1, ..., X)$.

The aggregated pile of hits $n[i], i \in \mathcal{X}$ generated by Algorithm 2 constitutes IPD as defined in Definition 16. In fact, this algorithm can be significantly simplified by mapping a whole OS instead of a fragment derived from it, thereby avoiding fragmentation in silico altogether (Algorithm 3). As the fragments of the same type (x, y) are totally uncorrelated since they almost surely come from different OSes of that type, it is justified to model their maps $\boldsymbol{\xi}^{xy}$ as $n_{(xy)}$ uncorrelated uniformly distributed random variables on $[x, y]$ instead.

$$\boldsymbol{\xi}^{xy} = (\xi_1, \xi_2, ..., \xi_{n_{(xy)}}) \sim \boldsymbol{\mathcal{U}}_{[x,y]}$$

Data: $N_s, p(x, y; \theta), w, \mathcal{X}$

Result: The total number of fragment mapping events versus the position on the sequence $n[i], i = 1, ..., X$

Initialise:

1. Under the assumption made above, the number of fragments of type (x, y) in the sample can be replaced by its mean value according to Eq. 4.3 and Eq. 4.6

$$n_{(xy)} = \frac{p(x, y; \theta)[(y - x + 1)/w]}{\sum_{(x,y)\in\mathcal{X}\times\mathcal{X}} p(x, y; \theta)[(y - x + 1)/w]} N_s$$

2. $n[i] = 0,\ i \in \mathcal{X}$

3. $j = 1$

for $(x, y) \in \mathcal{X} \times \mathcal{X}$ **do**

  for $j \leq n_{(xy)}$ **do**

    Generate a fragment of type (x, y) $(x', y') \in (x, y)$;

    Choose a random position inside this fragment $\xi \sim \mathcal{U}_{[x',y']}$;

    Assign the corresponding position on the reference sequence a hit

    $n[\xi] \coloneqq n[\xi] + 1$;

    $j \coloneqq j + 1$;

  end

end

Algorithm 2: Simulation of IPD

Note that the number of OSes to be mapped in Algorithm 3 should be artificially inflated to equal the number of fragments generated in Algorithm 2. The aggregated pile of hits $n[i], i \in \mathcal{X}$ constitutes IPD defined in Definition 16.

Data: $N_s, p(x, y; \theta), w, \mathfrak{X}$

Result: The total number of fragment mapping events versus the position on the sequence $n[i], i = 1, ..., X$

Initialise:

1. Generate $n_{(xy)}$ fragments, where the number of fragments of type (x, y) in the sample according to Eq. 4.3 and Eq. 4.6 is

$$n_{(xy)} = \frac{p(x, y; \theta)[(y - x + 1)/w]}{\sum_{(x,y)\in\mathfrak{X}\times\mathfrak{X}} p(x, y; \theta)[(y - x + 1)/w]} N_s$$

2. $n[i] = 0,\ i \in \mathfrak{X}$

for $(x, y) \in \mathfrak{X} \times \mathfrak{X}$ **do**
  Generate a random vector of $n_{(xy)}$ uniformly distributed random numbers on $[x, y]$
  $\boldsymbol{\xi} = \mathfrak{U}_{[x,y]};\ length(\boldsymbol{\xi}) = n_{(xy)};$
  Assign the corresponding positions on the reference sequence a hit
  for $j \in n_{(xy)}$ **do**
    $n[\boldsymbol{\xi}[j]] \coloneqq n[\boldsymbol{\xi}[j]] + 1$
  end
end

Algorithm 3: Simulation of IPD, simplified

4.4 Mapping frequency

4.4.1 Derivation of fragment frequency

The question posed in this section is how frequently an individual sequence element as compared to the other elements within the truncated sequence $\mathfrak{X}$ is expected to occur in IPD after mapping N_s randomly sampled fragments, herein

referred to as "mapping frequency". The mapping frequency will be used as a prior model in the subsequent statistical analysis of IPD.

Definition 17 (Mapping frequency)**.** *The mapping frequency $p(i)$ is the probability of a sequence position $i, i \in \mathcal{X}$ to receive a hit after a mapping of a fragment randomly drawn from the pool of fragments.*

The mapping frequency can be calculated using the frequency of fragments in the pool $p(x, y; \theta)$

$$p(i, \theta) = \frac{\sum_{(x,y)\in\mathcal{X}\times\mathcal{X}} p(x, y; \theta)\mathbf{1}_{i\in[x,y]}}{\sum_{i=1}^{X}\sum_{xy} p(x, y; \theta)\mathbf{1}_{i\in[x,y]}} = \frac{\sum_{x\leq i, x\in\mathcal{X}}\sum_{y\geq i, y\in\mathcal{X}} p(x, y; \theta)}{\sum_{i=1}^{X}\sum_{xy} p(x, y; \theta)\mathbf{1}_{i\in[x,y]}} \tag{4.9}$$

4.4.2 Derivation of mapping frequency $p(i, \theta)$

Here I derive the mapping frequency for a particular case described in Chapter 3.

Given Eq. 3.5

$$p(x, y|s = 2; \theta) = p(y|x, s = 2; \theta)p(x|s = 1; \theta)$$

and Eq. 4.9

$$p(i, \theta) = \frac{\sum_{xy} p(x, y; \theta)\mathbf{1}_{i\in[x,y]}}{\sum_{i=1}^{X}\sum_{xy} p(x, y; \theta)\mathbf{1}_{i\in[x,y]}} = \frac{\sum_{x\in\mathcal{X}, y\in\mathcal{X}} p(x, y; \theta)\mathbf{1}_{i\in[x,y]}}{\Xi}$$

$$= \frac{\sum_{x\leq i, x\in\mathcal{X}}\sum_{y\geq i, y\in\mathcal{X}} p(x, y; \theta)\mathbf{1}_{i\in[x,y]}}{\Xi}$$

The mapping frequency for a given $x = x_j^*$ is a marginal cumulative probability m.f. with respect to y

$$F(i, x_j^*|\theta) = p(y \geq i|x_j^*; \theta) = \sum_{y \geq i, y \in \mathcal{X}} p(y, x_j^*|2; \theta) = p(x_j^*|1; \theta) \sum_{y \geq i, y \in \mathcal{X}} p(y|x_j^*, 2; \theta)$$

Then, for all $x \in X^* = \{x_1^*, x_2^*, ..., x_k^*\}, x \leq y$

$$F(i, i \geq x_k^*|\theta) = p(x_k^* \leq i|\theta) = \sum_{j=1}^{k} F(i|x_j^*; \theta) = \sum_{j=1}^{k} p(x_j^*|1; \theta) \sum_{y \geq i, y \in \mathcal{X}} p(y|x_j^*, 2; \theta)$$

Finally, the mapping frequency is

$$p(i; \theta) = \frac{F(i; \theta)}{\sum_{i=1}^{X} F(i; \theta)} \tag{4.10}$$

4.5 Derivation of the hit count distribution in IPD

Using the mapping frequency derived in Eq. 4.9 as a model for assigning a hit to a position i and assuming total independence of individual mapping events, the hit count distribution across the truncated sequence $\mathcal{X} = (1, ..., X)$, $\boldsymbol{n} = (n_1, ..., n_i, ..., n_X), N_s = \sum_i n_i$ can be described by a Multinomial distribution as the fragment mapping events are independent

$$p(\boldsymbol{n}; \theta) = N_s! \prod_{i=1}^{X} p(i; \theta)^{n_i} / n_i! \tag{4.11}$$

where N_s is the total number of mapped fragments and n_i is the total number of hits mapped to position i.

4.6 Parameter inference from IPD

The likelihood function of IPD ($\boldsymbol{n}$) given the prior "mapping frequency" model (Eq. 4.9) $p(i;\theta)$ and the MN distribution of the data (Eq. 4.11) is

$$L(\boldsymbol{n};\theta) = N_s! \prod_{i=1}^{X} p(i;\theta)^{n_i}/n_i! \tag{4.12}$$

The log-likelihood of IPD is

$$\mathcal{L}(\boldsymbol{n};\theta) = \ln L(\boldsymbol{n};\theta) = \sum_{i\in\mathcal{X}} n_i \ln p(i,\theta) + \ln N_s! - \sum_{i\in\mathcal{X}} \log n_i! \tag{4.13}$$

In order to find the set of optimal parameters $\hat{\theta}$, the log-likelihood of the data has to be maximised with respect to θ.

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(\boldsymbol{n};\theta) \tag{4.14}$$

It is interesting to compare the inferred parameters with the parameters used to simulate IPD. The parameter estimates are likely to deviate to some extent from those used to simulate the data. The most obvious reasons for that could be the truncation of the state space $\mathcal{X}$, the limited sample size N_s or the size of fragmentation w. So, ideally, it is worth investigating how sensitive the parameter estimates are to these constraints.

4.7 Parameter inference from IPD

4.7.1 Simulation setup

Let the probability $p(x, y; \theta)$ over the types of OS be

$$p(x, y; \theta) = p(0, y; \theta) = \frac{(1-\theta)^{(y-\lambda)}\mathbf{1}_{y>\lambda}}{\sum_{y=1}^{X}(1-\theta)^{(y-\lambda)}\mathbf{1}_{y>\lambda}} \tag{4.15}$$

Here I set $x = 0$ in order to reduce the dimensionality of the state space for simplicity. The OSes described by this function all have their start position at 0 and end position at $y, y \in \mathcal{X}$. This is just a simple example of a distribution over absorbing states of some MC.

The mapping frequency for this model (Eq. 4.15) can be easily derived using Eq. 4.9

$$p(i; \theta) = \frac{\mathbf{1}_{(0,\lambda)} + \mathbf{1}_{(\lambda,X)}(1-\theta)^{(i-\lambda)}}{\Xi} \tag{4.16}$$

where Ξ is a normalisation constant.
Simulation setup:

- The state space is $\mathcal{X} = [1, ..., 2000]$.
- $\lambda = 300; \quad \theta = 0.002$.
- Sample size: $N_s = 10^6$. Remember that the sample size has to be large enough, e.g. the average number of hits per sequence position should be large, in this case it is $N_s/2000 = 500$, which is sufficiently large yielding an error of only $1/\sqrt{500} = 0.044 = 4.4\%$.
- Fragmentation size: $w = 1$.

4.7.2 Results of the simulation

Fig. 4.1 shows the results of the simulation: the grey curve represents the raw data and the pink curve - the smoothed data. After having simulated IPD using

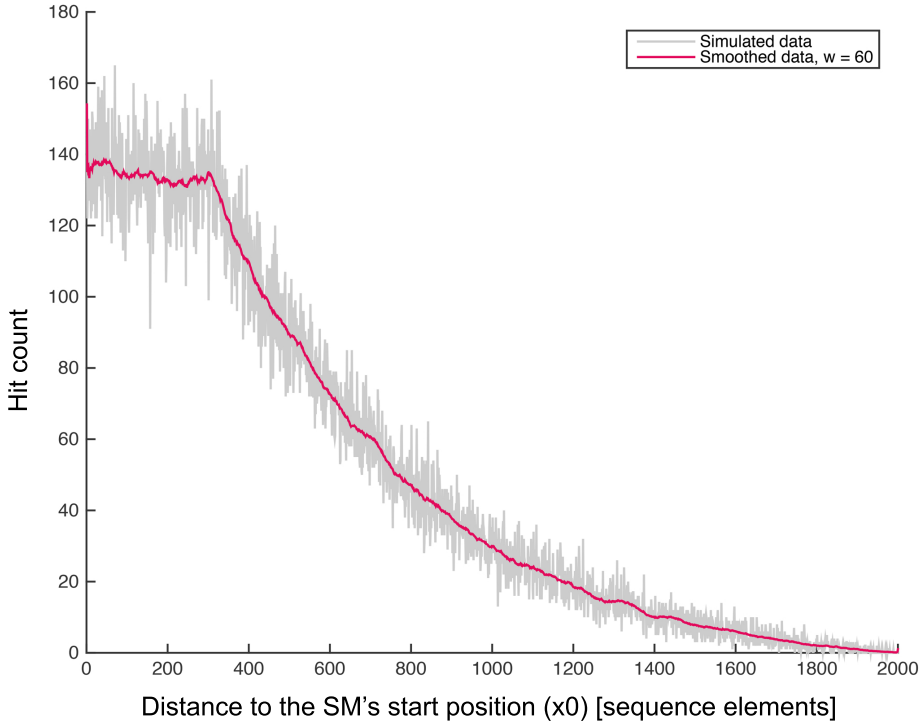


Figure 4.1. Simulated IPD in grey generated according to the model (Eq. 4.15) where $\theta_{synth} = 0.002$. Smoothed data - curve in pink. The smoothing window is 60. $N_s = 10^6$ and $w = 1$.

θ_{synth} and computed the log-likelihood function of IPD according to Eq. 4.13 and Eq. 4.16 I maximise it numerically to identify the optimal parameter $\hat{\theta}$ (Eq. 4.14). Provided the likelihood function has been derived correctly $\hat{\theta}$ is expected to be close to θ_{synth}.

Various numerical methods exist to find the maximum of the log-likelihood function, for example grid sampling or gradient descent or a combination of

both (see the preliminaries). Here I will use gradient descent initialising it with $\theta_0 = \theta_{synth}$, where θ_{synth} is the parameter used to simulate the data. In practice the log-likelihood function may have several local maxima, however gradient descent will only converge to one of those, sometimes arriving at a maximum which is only a local maximum rather than the global maximum. Ideally one should identify the interval of the parameter space which is the most likely to contain a global maximum (using grid sampling for instance) and then conduct a more detailed search in the predefined subspace using gradient descent. By

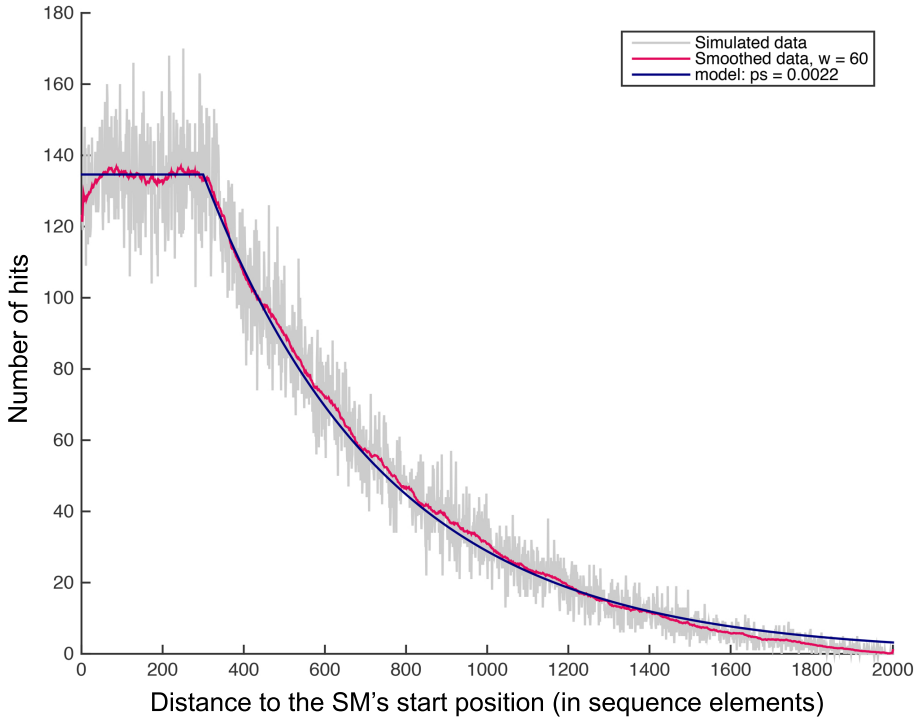


Figure 4.2. Simulated IPD in grey generated according to the model (Eq. 4.15) where $\theta = 0.002$. The smoothed curve is in pink. The smoothing window is 60. $N_s = 10^6$ and $w = 1$. The blue curve is the optimal model computed in 4.16 where $\hat{\theta}$ is substituted for θ, and where $\hat{\theta} = 0.0022$ was calculated by numerically maximising the log-likelihood function.

maximising the log-likelihood function of IPD simulated according to Algorithm

3 with $\theta_{synth} = 0.002$, $\mathcal{X} = [1...2000]$ and $w = 1$ the calculation arrives at the optimal parameter $\hat{\theta} = 0.0022$ which is very close to the original θ_{synth}. Also, LRT test (2.5.4) confirms that the model of mapping frequency (Eq. 4.16) fits the simulated IPD with at least 95% confidence.

Hence, we can conclude that the likelihood function derived in Eq. 4.12 is the right objective function to estimate the parameters of the MC on IPD generated according to Algorithm 3, e.g. we are sure to arrive at about the right estimate. The error of the estimate can be associated with the truncation of the state space $\mathcal{X}$ and insufficient sample size N_s. However, the error due to sample size should not exceed 4.4% on average in this particular case (as shown in Section 4.7.1). So, the main source of error must be due to the truncation of the state space. However, it does not affect the identification of the right model in this particular case and the model (Eq. 4.16) still fits the data.

Fig. 4.2 shows the graph of IPD together with the model curve (Eq. 4.16) computed for the optimal value of $\theta = \hat{\theta}$ and scaled to IPD's size by multiplying by the number of mappings (N_s) $p(i; \hat{\theta})N_s$.

4.8 Some analytical results

The aim of this section is to formally prove that the likelihood function of IPD derived in Eq. 4.12 reaches its maximum at the point in the parameter space used to generate these data ($\hat{\theta} = \theta_{synth}$) in the limit of infinitely large sample size $N_s \to \infty$.

4.8.1 Average mapping count

This section demonstrates some auxiliary lemmas useful to prove the subsequent key theorems of this chapter.

Consider multiple mappings of the fragment of type (a, b). Each mapping produces a random vector $\mathbb{1}_{\xi=\mathcal{U}_\delta} \coloneqq (00...1_\xi...00)$, $\delta = [a, b]$. After N mappings the cumulative vector $\boldsymbol{\eta} = \sum \eta$ approaches $N\mathbf{1}_\delta/[\delta]$ when N good to infinity

Lemma 4. *For* $\delta = [a, b], \eta = \mathbb{1}_{\xi=\mathcal{U}_\delta} \coloneqq (00...1_\xi...00)$ - *r.v. defined on* δ, $\mathbf{1}_\delta \coloneqq (00...1_a...1..1_b...00)$

$$\lim_{N\to\infty} \boldsymbol{\eta} = \lim_{N\to\infty} \sum_{i=1}^{N} \mathbb{1}_{\xi=\mathcal{U}_\delta} = N\mathbf{1}_\delta/[\delta]$$

Lemma 5. *For* $\delta = [a, b], \eta = \mathbb{1}_{\xi=\mathcal{U}_\delta}$

$$\sum_{i=1}^{N[\delta]} \eta \to N\mathbf{1}_\delta \; when \; N[\delta] \to \infty$$

Lemma 6. *For* $\delta = [a, b], \eta = \mathbb{1}_{\xi=\mathcal{U}_\delta}, \; w$

$$\sum_{i=1}^{N[\delta/w]} \eta \to N/w\mathbf{1}_\delta \; when \; N[\delta/w] \to \infty$$

4.8.2 Large sample size

Lemma 7. *If $p(x,y)$ is the concentration of fragments of type (x,y) in the pool, N - total size of the sample, $N \ll N_{pool}$, then*

$$N_{xy} \to p(x,y)N, \ when \ N \to \infty$$

Proof. The fragments are uncorrelated and the sample size N is small compared to the total pool size.
Then, the model of N draws with replacement with probability of success $p(x,y)$ is

$$p(N_{xy} = k) = \binom{N}{k} p(x,y)^k (1 - p(x,y))^{N-k}$$

$$\mathrm{E}(N_{xy}) = p(x,y)N$$

$$\mathrm{Var}(N_{xy}) = (1 - p(x,y))p(x,y)N \approx p(x,y)N$$

$$N_{xy} \approx p(x,y)N(1 \pm 1/\sqrt{p(x,y)N})$$

$$N_{xy} \approx p(x,y)N, \ when \ N \to \infty$$

□

4.8.3 Mathematical representation of IPD

Definition 18 (IPD(m))**.** *IPD can be mathematically defined as a vector - aggregate of random vectors $\eta = \mathbb{1}_{\xi=\mathcal{U}_\delta}$, $\boldsymbol{\eta} = \sum_{i=1}^{N} \eta$ after mapping of $N = \sum_{x,y} N_{xy}[(y-x+1)/w]$ uncorrelated(!) fragments. $\boldsymbol{\eta}$ is an X-dimensional vector.*

Theorem 10. *For $\delta = [a,b]$, $\eta = \mathbb{1}_{\xi=\mathcal{U}_\delta}$, $\boldsymbol{\eta} = \sum_{i=1}^{N} \eta$, $N = \sum_{x,y} N_{xy}[(y - x +$*

$1)/w]$ *and* $w = 1$

$$\boldsymbol{\eta} = \sum \eta \to N \sum_{(x\in\mathcal{X}, y\in\mathcal{X})} p(x,y)\mathbf{1}_{[x,y]},\ \textit{when}\ N \to \infty$$

Proof.

$$N = \sum_{x\in\mathcal{X}, y\in\mathcal{X}} N_{xy}[(y-x+1)/w]\ \&\ w = 1 \Rightarrow$$

$$\boldsymbol{\eta} = \sum_{i=1}^{N} \eta = \sum_{(x\in\mathcal{X}, y\in\mathcal{X})} \sum_{i=1}^{N_{xy}(y-x+1)} \eta \tag{4.17}$$

This approximation follows from Lemma 5:

$$\sum_{i=1}^{N_{xy}(y-x+1)} \eta \approx N_{xy}\mathbf{1}_{[x,y]} \tag{4.18}$$

This approximation is applicable when $n = N_{xy}(y-x+1) \gg 100$, since the error of the approximation scales as $1/\sqrt{n}$ (Lemma 4 and Lemma 5).
For example, $1/\sqrt{n} \ll 0.1$ when $n \gg 100$.
Then substituting Eq. 4.18 into Eq. 4.17 and using the result of Lemma 7 ($N_{xy} \approx p(x,y)N$, when $N \to \infty$) finally obtain

$$\boldsymbol{\eta} \approx \sum_{(x\in\mathcal{X}, y\in\mathcal{X})} N_{xy}\mathbf{1}_{[x,y]} \approx N \sum_{(x\in\mathcal{X}, y\in\mathcal{X})} p(x,y)\mathbf{1}_{[x,y]},\ \textit{when}\ N \to \infty$$

□

Next let us consider a generalised case where the fragmentation size is $w > 1$.

Theorem 11. *For* $\delta = [a, b]$, $\eta = \mathbb{1}_{\xi=\mathcal{U}_\delta}$, $\boldsymbol{\eta} = \sum_{i=1}^{N} \eta$, $w > 1$

$$\boldsymbol{\eta} = \sum \eta \to N/w \sum_{(x\in\mathcal{X}, y\in\mathcal{X})} p(x, y)\mathbf{1}_{[x,y]},\ when\ N \to \infty$$

Proof. The proof is almost identical to the one described in Theorem 10, with the only exception that instead of Lemma 5 use Lemma 6. □

Definition 19 (Normalised IPD)**.** *The normalised IPD is*

$$\boldsymbol{\zeta} = \frac{\boldsymbol{\eta}}{\sum \boldsymbol{\eta}}$$

Definition 20 (Parametric form of IPD)**.** *The parametric form of the normalised IPD is*

$$\boldsymbol{\eta} = \boldsymbol{\eta}(\theta) = \boldsymbol{\eta}(p(x, y; \theta), N)$$

$$\boldsymbol{\zeta}(\theta) = \frac{\boldsymbol{\eta}(\theta)}{\sum \boldsymbol{\eta}(\theta)}, i \in \mathcal{X}$$

Lemma 8. *The normalised IPD defined above can be approximated by the theoretical mapping frequency* $p(i, \theta)$ *when sample size* N *is very large*

$$\zeta_i \to p(i; \theta)\ when\ N \to \infty$$

Proof. According to the Theorem 10 the data vector can be approximated by

$$\boldsymbol{\eta} \approx N/w \sum_{(x\in\mathcal{X}, y\in\mathcal{X})} p(x, y; \theta_{synth})\mathbf{1}_{[x,y]}$$

provided the sample size N is sufficiently large

Then the projection of $\boldsymbol{\eta}$ on to the reference sequence $i \in (1...X)$ is

$$\eta_i \approx N/w \sum_{(x\in\mathcal{X},y\in\mathcal{X})} p(x,y;\theta_{synth})1_{i\in[x,y]}$$

Finally normalised IPD can be represented as

$$\zeta_i(\theta_{synth}) = \frac{\eta_i(\theta_{synth})}{\sum \eta_i(\theta_{synth})} \approx \frac{\sum_{(x\in\mathcal{X},y\in\mathcal{X})} p(x,y;\theta_{synth})1_{i\in[x,y]}}{\sum_i \sum_{(x\in\mathcal{X},y\in\mathcal{X})} p(x,y;\theta_{synth})1_{i\in[x,y]}} = p(i;\theta_{synth})$$

The rightmost hand side stems from Eq. 4.9

$$\zeta_i(\theta_{synth}) \approx p(i;\theta_{synth})$$

□

4.8.4 Inference from IPD

Theorem 12 (Inference from IPD)**.** *Log-likelihood function of IPD$\boldsymbol{\eta}(\theta_{synth})$ reaches its maximum at the point of the parameter space θ_{synth}, used to generate these data.*

Proof. IPD is multinomially distributed (Eq. 4.11 and the log-likelihood of MN distribution reaches its maximum when (Section 2.5.2)

$$\frac{n_i}{\sum n_i} = p(i;\hat{\theta})$$

where $p(i,\theta)$ is the probability of generating an i^{th} distant data point, n_i total number of data points at position i;
For IPD generated using parameter vector θ_{synth}

$$\zeta_i(\theta_{synth}) = \frac{\eta_i(\theta_{synth})}{\sum \eta_i(\theta_{synth})}$$

According to Lemma 8 in the limit of large sample sizes N $p(i;\theta_{synth})$ describes the limit behaviour of $\zeta_i(\theta_{synth})$

$$\zeta_i(\theta_{synth}) = \frac{\eta_i(\theta_{synth})}{\sum \eta_i(\theta_{synth})} = p(i;\hat{\theta}) \rightarrow p(i;\theta_{synth}) \; when \; N \rightarrow \infty$$

So, in the limit of large N, θ used to generate IPD indeed maximises its likelihood

$$\lim_{N\rightarrow\infty} \hat{\theta} = \theta_{synth}$$

□

Chapter 5

Processing of real pileup sequencing data (RPD)

In the previous chapter the case of "ideal" pileup data (IPD) was considered - an idealisation of pileup sequencing data where certain constraints are relaxed. IPD were defined as population scale mappings of the fragments obtained upon fragmentation of the sequence segment outputs (OS) of large population of SMs. The number of fragments mapped was assumed to be very large $N_s \to \infty$, yet the pool of sequence segments generated by a population of SMs was assumed to be even several orders of magnitude larger $N_{pool} \gg N_s$ to minimise the chance of two random fragments originating from the same sequence segment. Each fragment was mapped to the reference sequence by choosing a random position inside it and assigning a hit to the corresponding position on the reference sequence. The multinomial distribution proved a good approximation of IPD and the maximum log-likelihood allowed us to closely estimate the values of the parameters used to generate SM's MC. In this chapter I will clarify the context in which Real Pileup

Data (RPD) have to be considered and establish the applicability of the analysis tool developed previously in light of the details of the real experiment.

5.1 Clarification of pileup data acquisition

5.1.1 Background fragments

Selection of specific fragments (which carry the protein of interest) is a multi-step process including antibody binding, magnetic bead attaching to the antibody, and physical isolation of magnetic beads. In theory, we should expect to see only specific fragments in the sample, as the antibodies used in ChIP are specifically designed to select only these fragments. However, other random DNA fragments may occasionally piggyback on the specific fragments at some point in ChIP. Although the mechanism of how random fragments end up in the sample is poorly understood, one reasonable way of quantifying IP efficiency would be to calculate the ratio of the mean signal read count to the background read count (Bao *et al.*, 2013).

Since the occurrence of background fragments should not be sequence dependent we expect that upon mapping they make uniform contribution to the distribution of hit counts over the entire span of the chromosome.

5.1.2 Mapping

In contrast to mapping of a random position inside the fragment featured in the previous chapter, in this case it is the 5' end k contiguous nucleotides that are mapped to the reference sequence and each successful mapping ends up

contributing hits to RPD along the k-long segment (Fig. 5.1 and Chapter 2). These hits will be later referred to as "mapping".

5.1.3 Small fragment sample, dilute sample

In the actual ChIP-Seq experiment the DNA extracted from the population of cells gets fragmented into fragments of variable size $\sim 150 - 500$ nt (Chapter 2, Fig. 5.3). Hence, if each cell donates at least one chromosome to be fragmented the total number of fragments in the pool becomes approximately $N_{pool} \approx SL/w$, where $S \sim 10^8$ is the size of the bacterial colony and $L \sim 10^6$ is the length of a single chromosome and $w \sim 10^2$ - the order of fragment length.

$$N_{pool} \approx SL/w \sim 10^8 * 10^6/10^2 = 10^{12}$$

Only a sample of fragments (containing both protein bound fragments and background fragments) is isolated from the pool for further analysis (N_s). The size of the sample is dependent on the experimental conditions, yet it is always several orders of magnitude smaller than the size of the initial fragment pool ($N_s \ll 10^{12}$). The recommended number of uniquely mapped reads to ensure an optimal read coverage is ~ 8 millions for a fruit fly with a genome of 130 Mbp (Furey, 2012). If this recommendation is extended to the genome of *E.coli* with 4 Mbp only about $2 \cdot 10^5$ reads should be sufficient for reliable detection of protein enrichment. This means only a tiny proportion of all fragments (sample) would be selected for sequencing.

The small sample size satisfies the requirement of dilute sample ($10^5 \ll 10^{12}$) used to derive the results in Chapter 4. Permitting only a dilute sample in the model allows to entirely disregard the chance of any two fragments originating from the

same chromosome and consider the sampled fragments drawn independently and with replacement from the pool.

With the requirement of dilute sample being satisfied the model derived in Chapter 4 still relies on another major assumption - high density of hits (infinite sample size - $N_s \to \infty$). Given 10^5 mapped reads RPD would contain only $5 \cdot 10^6$ hits (given tag length of 50) scattered over 4 million genomic positions generating a very noisy signal with a low hit density.

5.1.4 Sequencing bias

The composition of sampled fragments is registered by sequencing (reading sequence base pairs) before they can be quantified and this registration process is always associated with some form of bias.

Prior to sequencing the fragments are amplified meaning their number gets artificially inflated. PCR amplification is usually uneven and can introduce some bias into fragment quantification. For example fragments containing more GC base pairs tend to get amplified to a greater extent than those containing less GC (Goren *et al.*, 2010; Kozarewa *et al.*, 2009). To reduce sequencing bias introduced by PCR read duplicates are often removed prior to mapping because they are likely to result from such amplification. The likelihood of any two reads representing twice the same fragment becomes even higher once we take into account possible sequencing errors. Appearance of the same error in any two sequenced fragments makes it very unlikely that they have been sampled independently. These two read copies if quantified independently would hamper the registration of the true fragments present in the sample.

However, when the sequencing error is low ($< 0.33\%$ per base pair = 1 error per

300 bp fragment) identical reads may simply represent independently sampled identical fragments originating from a genomic area where the protein of interest binds very frequently. Identical fragments may be sampled from the pool simply by chance merging with the true PCR duplicates. Therefore, discarding such reads would distort the representation of relative frequency of binding necessary to explore the model of the DNA binding protein activity.

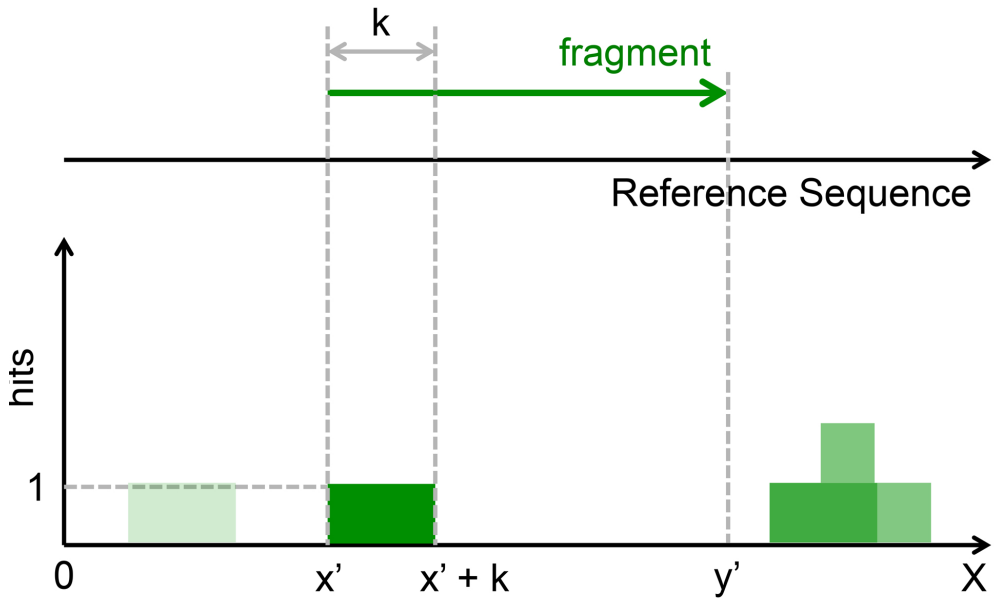


Figure 5.1. This diagram shows a match between a fragment (x', y') and the Reference sequence. The mapping (green step) of k first elements of a fragment contributes to assigning hits $(+1)$ along $(x', x' + k)$ segment of the sequence. Lighter green steps correspond to earlier mappings.

5.2 Simulation of RPD

5.2.1 Introducing background to the initial pool of OS

Like in the previous chapter consider sequence segments (OS) of all types $(x, y), x \in \mathcal{X}, y \in \mathcal{X}$ produced by a population of SMs at the end of their life cycle. Let us also introduce a constant background component to this population of OSes. After fragmentation the sample will be drawn from the combined pool of fragments where both OS fragments and background fragments are present.

$$N_{pool} = N_{OS} + N_b$$

$$f_b = \frac{N_b}{N_{pool}} = const$$

Note that the pool of fragments must be significantly larger than the pool of OS fragments

$$N_{OS} \ll N_b$$

This is because SM interacts only with a small proportion of the population of sequences, yet the rest also gets fragmented together with the OSes. Background segments are assigned a type $(1, X)$, which corresponds to the full sequence span of interest $(\mathcal{X} = (1, ..., X))$.

5.2.2 Fragmentation of the pool

Each sequence segment in the pool is subjected to fragmentation into small fragments, their size being normally distributed with mean length $\mu_w = 200$

and variance $\sigma_w = 30$ (Fig. 5.3).

$$\{\boldsymbol{\xi} = (\xi_-, \xi_+)_i \in (x, y), (x, y) \in \mathfrak{X} \times \mathfrak{X} \cup (0, X)\},\ i \in N$$

5.2.3 A fragment draw from the pool

Using Bayes formula the probability of picking up a fragment that belongs to the OS fraction of fragments (signal)

$$p(OS|draw) = \frac{p(draw, OS)}{p(draw)} =$$

$$\frac{p(draw|OS)p(OS)}{p(draw|OS)p(OS) + p(draw|background)p(background)} =$$

$p(OS)$ is the probability of finding a fragment of OS type in the pool, which is the same as its fraction in the pool

$$p(OS) = f_{OS}$$

accordingly

$$p(background) = f_b$$

$p(draw|OS)$ is the probability of actually drawing the fragment of OS type that has been identified

$$p(draw|OS) \coloneqq p_{OS}$$

accordingly

$$p(draw|background) \coloneqq p_b$$

Let us also demand that when making a random draw from the pool, the background-type fragments are less likely to be picked up than the OS-type

fragments

$$p(draw|OS) = p_{OS} > p(draw|background) = p_b$$

That is because the "draw" must be biased towards the OS fragments, otherwise the meaningful proportion of the fragments (signal) would be masked by the background junk, which is abundantly present in the initial pool (Section 5.2.1).

Then $p(OS|draw)$ can be expressed as a function of these parameters:

$$p(OS|draw) = \frac{p_{OS}f_{OS}}{p_{OS}f_{OS} + p_b f_b} \tag{5.1}$$

Now, the probability that the outcome of the draw is a fragment that belongs to the background can be expressed in a similar way using Bayes formula

$$p(b|draw) = \frac{p_b f_b}{p_{OS}f_{OS} + p_b f_b} \tag{5.2}$$

Let us divide both the numerator and denominator by a constant $p_{OS}f_{OS}$ ($f_{OS} = 1 - f_b = const$, see Section 5.2.1) and introduce a new constant $C = p_b f_b / p_{OS} f_{OS}$. Note that constant C is proportional to the probability of background draw p_b. This constant can also be seen as the specificity of the selector (that makes the draw) to the background-type fragments.

$$C = \frac{p_b f_b}{p_{OS}f_{OS}} = \frac{p_b f_b}{p_{OS}(1 - f_b)} \propto p_b$$

So it is a convenient measure of the effect of the background fragments on the sample.

Now, using the new notation the probability of drawing an OS fragment can be rewritten as

$$p(OS|draw) = \frac{1}{1 + C} \tag{5.3}$$

and the background-fragment

$$p(background|draw) = \frac{C}{1+C} \tag{5.4}$$

As for drawing a particular fragment of type (x, y), it follows from Eq. 5.1, Eq. 5.3 and $\sum_{(x,y)\in\mathfrak{X}\times\mathfrak{X}} f_{xy} = f_{OS}$, where f_{xy} is the fraction of the fragment of type (x, y) in the pool, that

$$p((x, y)_f|draw) = \frac{f_{xy}/f_{OS}}{1+C} \tag{5.5}$$

$(x, y)_f$ - fragment of type (x, y).

After fragmentation an OS of type (x, y) is expected to donate $[(y - x + 1)/\mu_w]$ fragments to the pool on average. So, the expected number of fragments of type (x, y) in the pool is

$$\mathrm{E}(n_{(xy)}) = N_{pool}p(x, y)[(y - x + 1)/\mu_w]$$

Hence, the frequency of an (x, y)-OS-type fragment is

$$f_{xy} = f_{OS}\frac{p(x, y)[(y - x + 1)/\mu_w]}{\sum_{x\in\mathfrak{X},y\in\mathfrak{X}} p(x, y)[(y - x + 1)/\mu_w]} \tag{5.6}$$

Substituting Eq. 5.6 into Eq. 5.5 the probability of drawing an OS fragment of type (x, y) is

$$p((x, y)_f|draw) = \frac{1}{1+C}\frac{p(x, y)[(y - x + 1)/\mu_w]}{\sum_{x\in\mathfrak{X},y\in\mathfrak{X}} p(x, y)[(y - x + 1)/\mu_w]} \tag{5.7}$$

5.2.4 Fragment sampling from the pool

The next step is to isolate a small sample of fragments (sample) from the pool $N_s = n_{OS} + n_b$, where n_{OS} is the total number of fragments that belong to the

OS component of the pool

$$n_{OS} = \sum_{(x,y)\in\mathcal{X}\times\mathcal{X}} n_{(x,y)}$$

and n_b is the number of fragments of background type in the sample.

Due to the small sample size N_s, the distributions $n_{(xy)}, n_b$ of the fragments over the types $\{(x,y), background\}$ can be modelled using a multinomial distribution with the probabilities of drawing individual fragments derived in Eq. 5.4 (background) and Eq. 5.7 (OS).

A small sample size warrants that the previous draws do not affect the overall composition of the pool, hence they do not change the probabilities of the outcomes of the consequent draws. Therefore, the model of draws from different types with replacement is applicable in this case.

$$(\boldsymbol{n}_{OS}, n_b) \sim MN(N_s, \boldsymbol{p}(OS|draw), p(background|draw)) \tag{5.8}$$

where $\boldsymbol{p}(OS|draw) \coloneqq (p((x,y)_f|draw), (x,y) \in \mathcal{X}\times\mathcal{X})$
and $\boldsymbol{n}_{OS} \coloneqq (n_{(xy)}, (x,y) \in \mathcal{X}\times\mathcal{X})$

As opposed to the case described in Chapter 4 where the data were simulated assuming $n_{(xy)}$ is very large, $n_{(xy)}$ might be realistically quite small $p(n_{(xy)} > 1) \ll 1$, which makes the approximation by average invalid in this case (unlike in the previous chapter).

A small sample size ($N_s \ll N_{pool}$) also warrants that no two fragments should originate from fragmentation of the exact same OS, which assures that the fragments drawn from the pool that belong to the same type are totally uncorrelated and random variables $\boldsymbol{\xi}_i^{xy}, i = 1, ..., n_{(xy)}$ are totally independent.

5.2.5 The model of fragmentation

Let us redefine fragment r.v. as $\boldsymbol{\xi} = (\xi, \Delta)$ where ξ is the middle of the fragment and Δ is its span, so $\boldsymbol{\xi}$ can be expressed as

$$\boldsymbol{\xi} = (\xi - \Delta, \xi + \Delta) = (\xi_-, \xi_+)$$

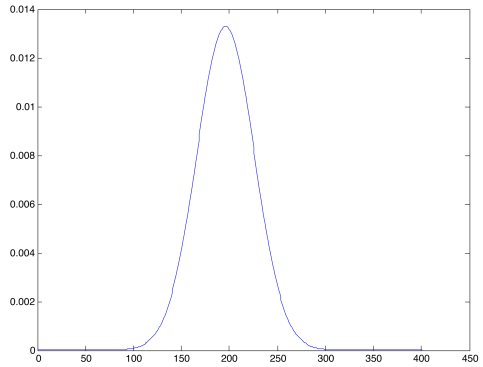
Then, for each fragment in the sample its location within the segment can be represented as a duplex of two random variables ξ and Δ. These random variables will signify the start and end positions of the simulated fragment

$$(\xi_-, \xi_+) = (\xi - \Delta, \xi + \Delta)$$

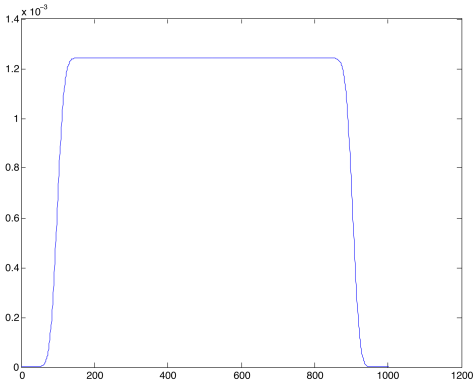
Δ is normally distributed $2\Delta \sim \mathcal{N}(\mu, \sigma)$. In order to obtain this distribution I utilised the distribution of the fragment size in the preparation library for ChIP-Seq (Fig. 5.3) after having subtracted the size of the adapters 83 from the mean of the library distribution in order to determine the distribution of the pure fragments. I also estimated the variance approximately from the graph ($\sigma \approx 30$). Note that though this is a typical fragment library the distribution of the fragment size should vary across the libraries of fragments in other protocols.

Given Δ, the distribution of ξ is uniform on $[x + \Delta, y - \Delta]$ for a fragment of type (x, y), the segment from which ξ is to be chosen is constrained by Δ because the centre of the fragment cannot be chosen too close to x or y:

$$\xi - x \geq \Delta \;\&\; y - \xi \geq \Delta$$



(a) The model of the probability distribution of the span of the fragment $p(2\Delta) = \mathcal{N}(278 - 83, 30)$, derived from Fig. 5.3.



(b) Distribution of the middle of the fragment $p(\xi|\Delta)$ conditional upon the chosen width of the fragment Δ (Eq. 5.9).

Figure 5.2. Fragmentation model derived from Fig. 5.3. Note that though this is a typical fragment library the distribution of the fragment size should vary across the libraries of fragments in other protocols.

$$p(\xi|\Delta) = \begin{cases} p(\xi, \xi < x + \Delta) = 0 \\ p(\xi, \xi > y - \Delta) = 0 \\ p(\xi, x + \Delta \leq \xi \leq y - \Delta) = 1/(y - x - 2\Delta) \end{cases}$$

Unconditional distribution of ξ $p(\xi)$ can be computed by integrating out Δ

$$p(\xi) = \int_{\Delta} p(\xi|\Delta)p(\Delta) = \frac{1}{\Xi}\Big[Erf\Big(\frac{\xi - \mu}{\sqrt{2}\sigma}\Big) - Erf\Big(\frac{\xi - (y - x - \mu + 1)}{\sqrt{2}\sigma}\Big)\Big] \quad (5.9)$$

where Ξ is the normalisation constant.

Fig. 5.2b shows the distribution of the position of the middle of a fragment obtained from Eq. 5.9. Here the initial unfragmented segment is of type $(0, 1000)$ and 2Δ follows $\mathcal{N}(195, 30)$ (following Fig. 5.2a).

Figure 1.1: Bioanalyzer traces of final library.

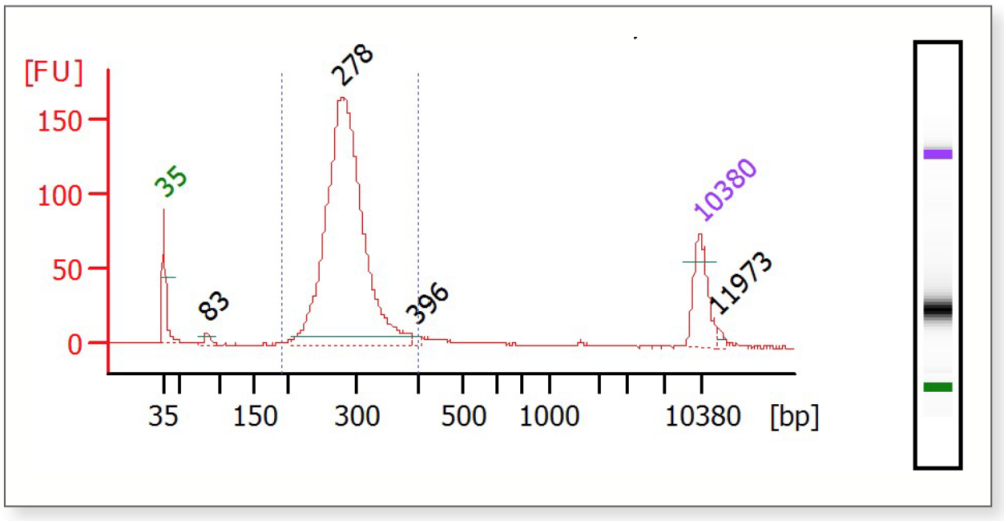


Figure 5.3. Bioanalyzer traces of final fragment library (after reverse cross-linking) prepared using NEBNext ChIP-Seq Library Prep Master Mix Set for Illumina. The graph shows the presence of fragments of various lengths in the library. The peaks correspond to the largest sub-populations of the fragments. Only 300 bp-fraction of the fragment library (separated from the rest with dotted lines) is selected for the next step in ChIP-Seq pipeline (sequencing). Reprinted from New England Biolabs, Inc. (Version 6.0).

5.2.6 Mapping

When mapping single stranded fragments to the reference sequence the hits are assigned to the first k elements of the corresponding sequence map

$$n[\xi_-, \xi_- + k] := n[\xi_-, \xi_- + k] + 1$$

Fragmentation of double stranded DNA generates fragments which can map to both strands of DNA, for instance background fragments. Such fragments would

generate k-long mappings at both ends with equal probability.
Such double-ended mappings can be modelled as

- assign hits to the leftmost k elements of the sequence map when $r = \mathcal{U}_{[0,1]} < 0.5$
- assign hits to the rightmost k elements of the sequence map when $r = \mathcal{U}_{[0,1]} \geq 0.5$

5.3 Simulation of real pileup data (RPD) of a single sequence segment

As an example, let us first simulate one type of sequence segment $(0, Y), Y = 1000$ (Algorithm 4). The results of the simulation are shown on Fig. 5.4 and Fig. 5.5 for double sided and single sided mappings respectively.

5.4 Simulation of the full RPD including the background

The aggregated pile of hits $n[i], i \in \mathcal{X}$ generated according to Algorithm 7 constitutes RPD.

Data: $N, k, p(2\Delta) = \mathcal{N}(\mu, \sigma), p(\xi|\Delta)$

Result: Pileup of hits: $n[i], i \in \mathcal{X}$

Initialise: $n[i] = 0,\ i \in \mathcal{X}$;

j= 0;

for $j \in N$ **do**

Draw a random number Δ, $\Delta \sim 1/2\ round(\mathcal{N}(\mu, \sigma))$;

Draw a random number ξ conditional upon the choice of Δ;

$\xi \sim \mathcal{U}_{[x+\Delta, y-\Delta]}$;

$\xi_- = \xi - \Delta$;

$\xi_+ = \xi + \Delta$;

$n[\xi_-, \xi_- + k] := n[\xi_-, \xi_- + k] + 1$

end

Algorithm 4: Simulation of RPD of a single type of OS

5.5 Model of mapping frequency

Here I slightly modify the proposed distribution over types $p(x, y; \theta)$ (Eq. 4.15)

$$p(x, y; \theta) = p(x_0, y; \theta) = \frac{(1-\theta)^{(y-x_0-\lambda)} \mathbf{1}_{y>\lambda+x_0}}{\sum_{y=1}^{X} (1-\theta)^{(y-x_0-\lambda)} \mathbf{1}_{y>\lambda+x_0}} \tag{5.10}$$

Then the frequency of mapping can be calculated for this distribution analogous to Eq. 4.16

$$p(i; \theta) = \frac{\mathbf{1}_{x_0 \le i \le x_0+a} + \mathbf{1}_{i>x_0+a}(1-p_s)^{(x-x_0-a)}}{\Xi} \tag{5.11}$$

where $[\theta, \lambda] := [p_s, a]$ and Ξ is the normalisation constant.

Then update $p(i; \theta)$ from Eq. 5.11 taking into account the background fraction of the mapping events

$$p(i; \theta) := \frac{p(i; \theta) + C/X}{\sum_{i=1}^{X} p(i; \theta) + C} \tag{5.12}$$

Data: $N, k, p(2\Delta) = \mathcal{N}(\mu, \sigma), p(\xi|\Delta)$

Result: Pileup of hits: $n[i], i \in \mathcal{X}$

Initialise: $n[i] = 0, i \in \mathcal{X}$;

j= 0;

for $j \in N$ **do**

  Draw a random number Δ, $\Delta \sim 1/2\ round(\mathcal{N}(\mu, \sigma))$;

  Draw a random number ξ conditional upon the choice of Δ;

  $\xi \sim \mathcal{U}_{[\Delta, X-\Delta]}$;

  $\xi_- = \xi - \Delta$;

  $\xi_+ = \xi + \Delta$;

  if $rand(1) \leq 0.5$ **then**

    $n[\xi_-, \xi_- + k] \coloneqq n[\xi_-, \xi_- + k] + 1$

  else

    $n[\xi_+ - k, \xi_+] \coloneqq n[\xi_+ - k, \xi_+] + 1$

  end

end

Algorithm 6: Simulation of RPD of the background

5.6 Results

Fig. 5.6 shows the results of the full simulation. The model described by Eq. 5.12 is a good approximation of these simulated RPD (fit with 95% confidence). The error mostly arises from the delay of the signal onset compared to the model due to the significant fragment size $\mu \approx 200$. Since the scale of the model is much larger than the size of the delay, fragmentation does not drastically affect the quality of fit provided the fragment size is reasonably small. The effect of the size of the fragment will be investigated in the next section.

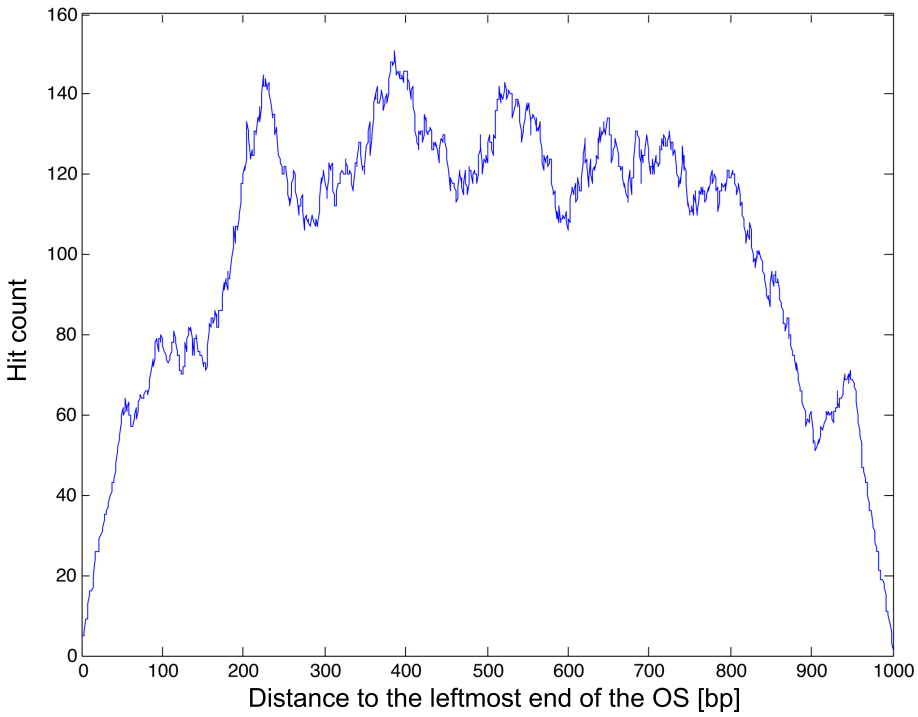


Figure 5.4. The pileup of a single OS of length 1K generated using double end fragment mapping. Both the first and last $k = 50$ elements of each fragment are mapped to the reference sequence. 0 is the leftmost and 1000 is the rightmost coordinate of OS on the reference sequence. The total number of fragments mapped is $N_s = 1000$.

5.6.1 Parameter inference (MLE)

Like in Chapter 4, the model of the frequency of mapping of a sequence position which was computed in Eq. 5.12 is now to be incorporated into the multinomial distribution of the hit count (Eq. 4.11) to compute the likelihood function and then to infer the optimal parameters of the model θ by maximising the log-likelihood.

First, let us calculate the log-likelihood for the simulated hit count $n = n_i, i =$

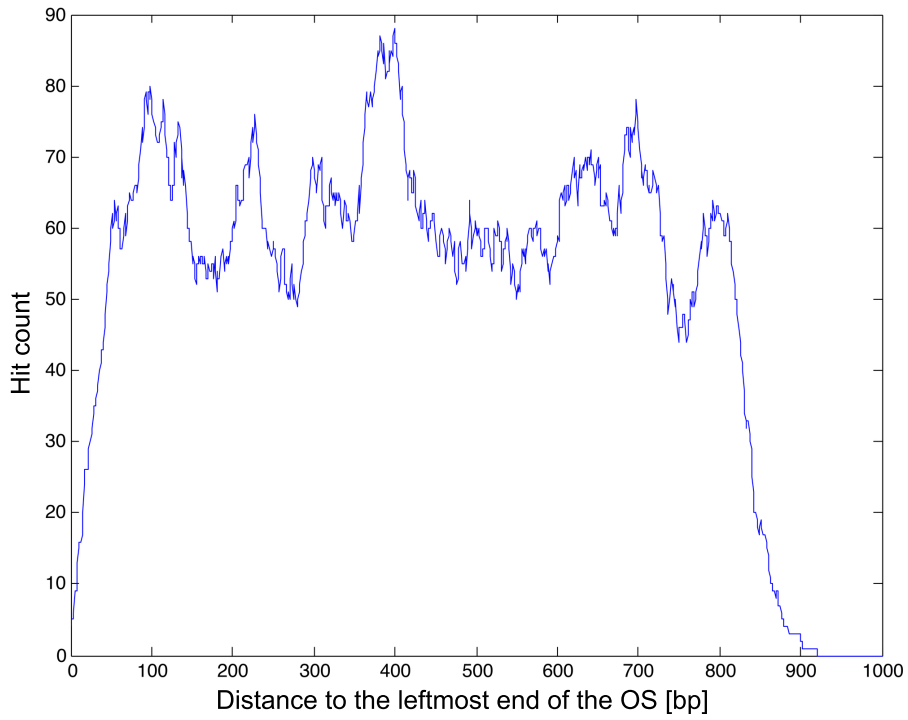


Figure 5.5. The pileup of a single OS of length 1K generated using double end fragment mapping. The first $k = 50$ elements of each fragment are mapped to the reference sequence. 0 is the leftmost and 1000 is the rightmost coordinate of OS on the reference sequence. The total number of fragments mapped is $N_s = 1000$.

$1, ..., X$

$$\mathcal{L}(\boldsymbol{n}; \theta) = \ln L(\boldsymbol{n}; \theta) = \sum_{i=1}^{X} n_i \ln p_i + const \tag{5.13}$$

According to 4.14

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(\boldsymbol{n}; \theta)$$

Data: $N_s, p(x, y; \theta), C, w, \mathcal{X} = [0, 1, ..., X]$

Result: Pileup data of the number of fragment mapping events vs. the position on the sequence $n[i], i \in \mathcal{X}$

Initialise:

1. Simulate $n_{(xy)}$ and n_b using a multinomial distribution (Eq. 5.8)
   $(\boldsymbol{n}_{OS}, n_b) \sim MN(N_s, \boldsymbol{p}(OS|draw), p(background|draw))$
   where $N_s = \sum_{(x,y)\in\mathcal{X}\times\mathcal{X}} n_{(xy)} + n_b$ is the total number of fragments.
2. $n[i] = 0,\ i \in \mathcal{X}$

for $(x, y) \in \mathcal{X} \times \mathcal{X}$ **do**
  Algorithm 4 ($n_{(xy)}$);
end
Algorithm 6 (n_b);

Algorithm 7: Algorithm of simulation of RPD

Parameter inference from the binned data

After grouping the data into bins of size W the new data is

$$n_j^* = \sum_{i=(j-1)*W+1}^{Wj} n_i$$

$$\boldsymbol{n}^* = \{n_j\}, j = 1, ..., X/W$$

We can now approximate the log-likelihood of the binned data $\boldsymbol{n}^*$ by least squares, since the noise becomes Gaussian-like after collapsing a large number of hits into a single bin (since the sample mean is normally distributed). In this case max-log-likelihood turns into least squares

$$\mathcal{L}(\boldsymbol{n}^*) = \sum_{i=1}^{X/W} \frac{(n_j^* - p(j;\theta)n^*)^2}{p(j;\theta)n^*} \tag{5.14}$$

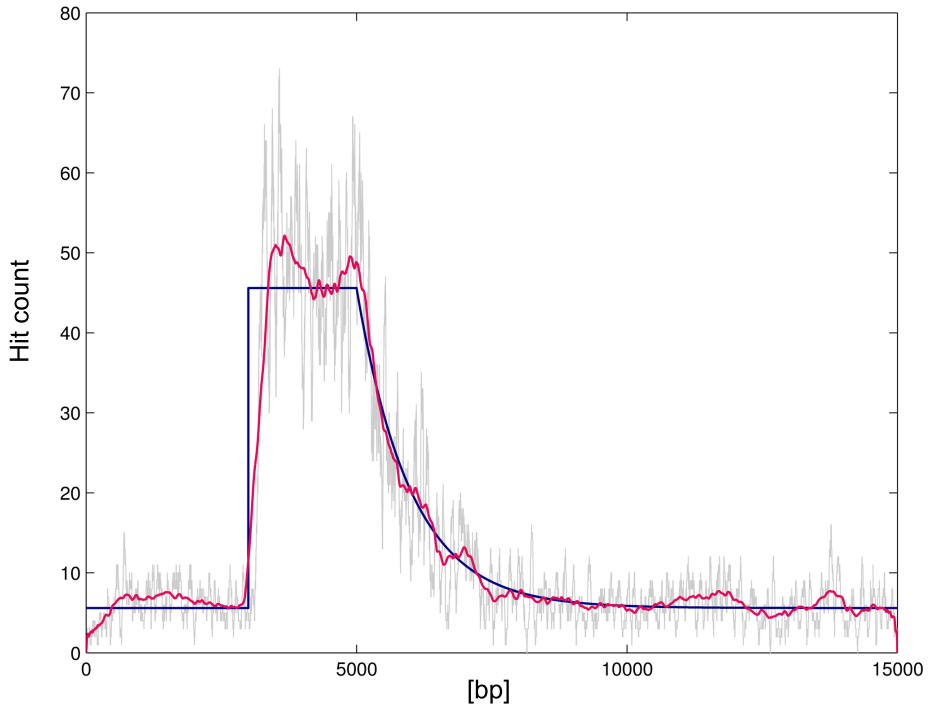


Figure 5.6. Grey - the simulated distribution of the hit count ($X = 15000; x_0 = 3000; N_s = 2000; C = 0.7; k = 50; \mu = 200, \sigma = 30$); Blue - function like in Eq. 5.12 ($p_s = 10^{-3}$), pink - smooth data ($\Delta = 500$).

where $n^* = \sum_j n_j^*$.

$p(j;\theta)$ is the scaled version of $p(i;\theta)$

$$\theta_W = \theta/W$$

$$X_W = X/W$$

The next step is to maximise Eq. 5.13 or minimise Eq. 5.14 to infer the optimal parameters $\hat{\theta}_W$

$$\hat{\theta} = W\hat{\theta}_W$$

5.7 Parameter sensitivity

5.7.1 Effect of sample size

When sample size N_s increases the parameter estimate converges to its true value when estimated on the simulated data, provided each fragment contains a single element ($\mu = 1$) and the mapping size is $k = 1$. The error of the estimate scales as $1/\sqrt{N_s}$ and goes to zero when the number of fragments goes to infinity (Fig. 5.7).

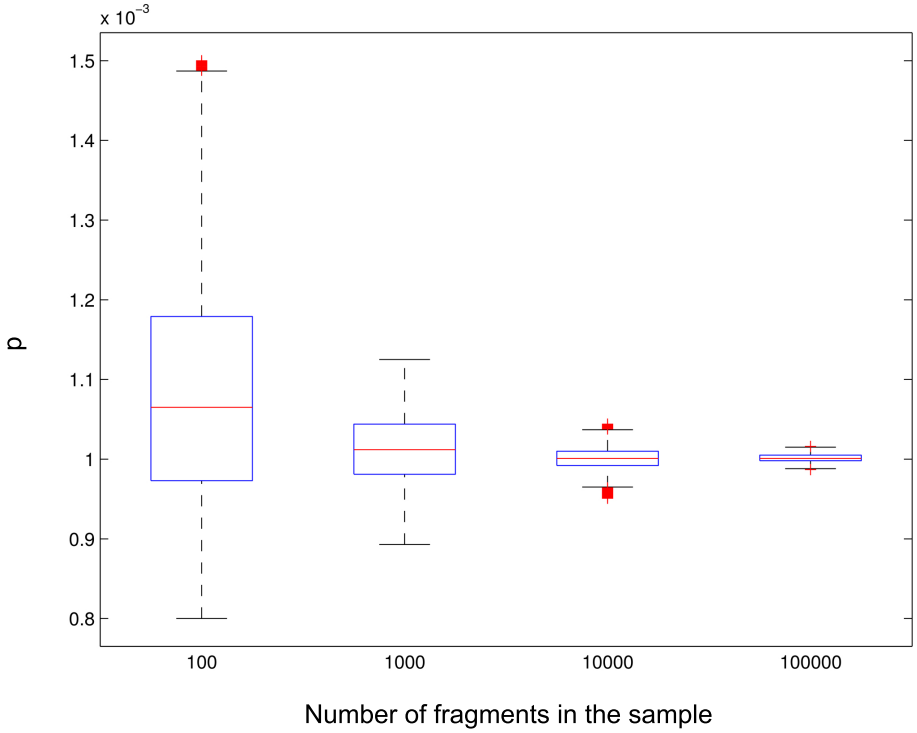


Figure 5.7. Distribution of optimal estimates of p_s (ideal $p_s = 10^{-3}$) for different sample sizes $N_s = 10^2, 10^3, 10^4, 10^5$, other parameters: $C = 0,\ a = 2000,\ \mu = 1,\ \sigma = 0,\ k = 1.$

5.7.2 Effect of average fragment size

When introducing a finite fragment size ($\mu \gg 1$) the optimal parameter estimated on the simulated data shifts away from its true value (Fig. 5.8). This means a finite fragment size introduces a systematic error in the parameter estimation. The error increases with the fragment size.

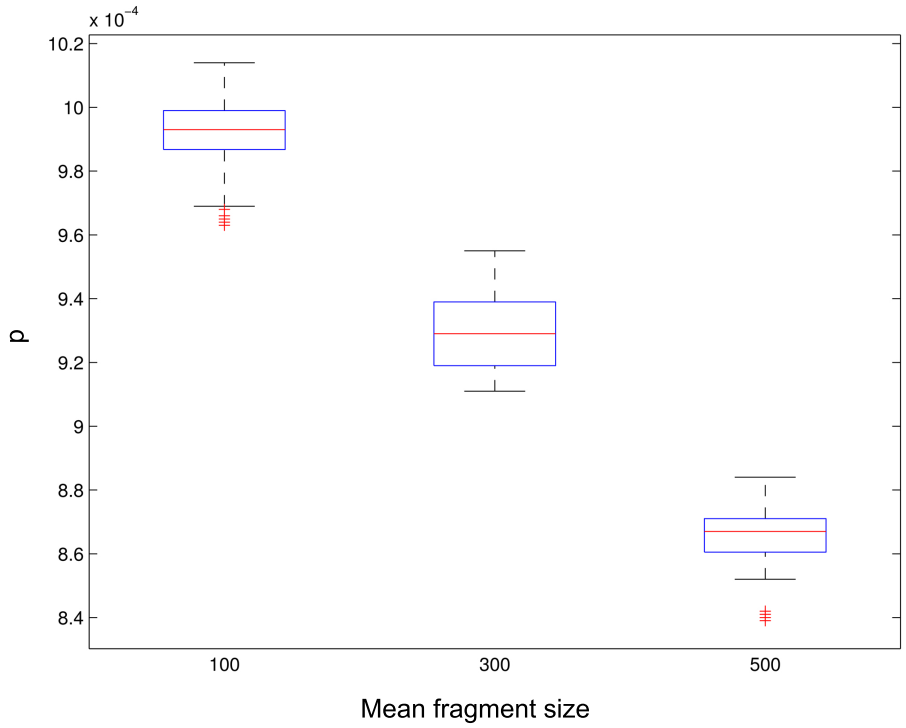


Figure 5.8. Distribution of optimal estimates of p_s (ideal $p_s = 10^{-3}$) for different fragment average sizes $\mu = 100, 300, 500$, sample size $N_s = 10^4$ other parameters: $C = 0,\ a = 2000,\ \sigma = 0,\ k = 50$.

5.7.3 Effect of fragment size variability

With increased fragment size variability σ, it is the estimate spread that slightly increases, the systematic error remaining unchanged (Fig. 5.9).

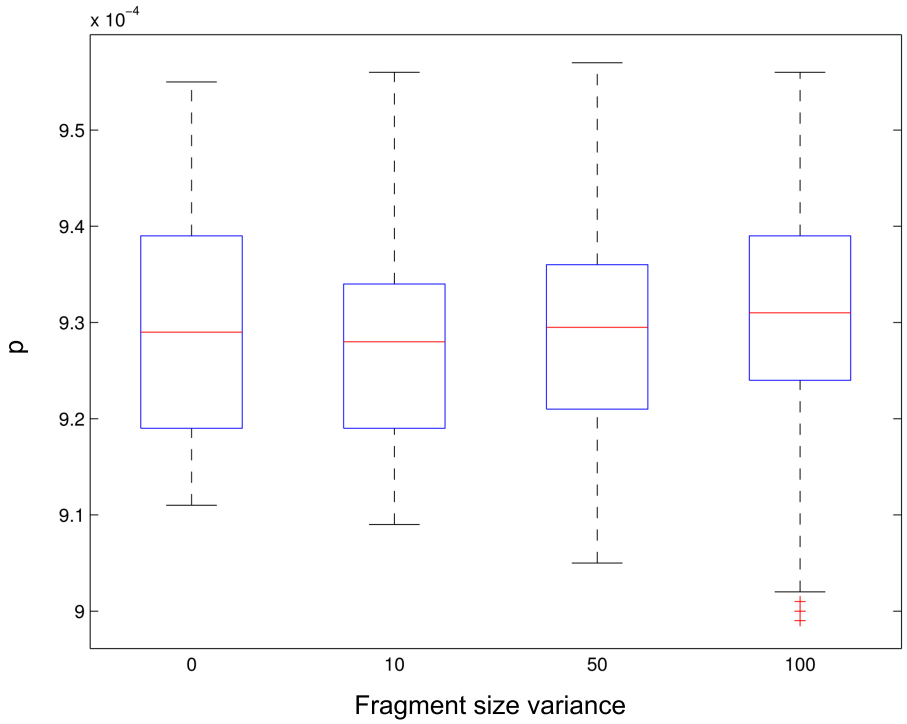


Figure 5.9. Distribution of optimal estimates of p_s (ideal $p_s = 10^{-3}$) for different standard deviations of the fragment size: $\sigma = 0, 10, 50, 100$, sample size $N_s = 10^4$, other parameters: $C = 0,\ a = 2000,\ \mu = 300,\ k = 50$.

5.7.4 Effect of background

Both systematic and random errors (Fig. 5.10) increase as the background level increases. The level of the background is mathematically represented by a constant C which is proportional to ChIP's specificity to the background-type

fragments.

$$C = \frac{p_b f_b}{p_{OS}(1 - f_b)} \propto p_b$$

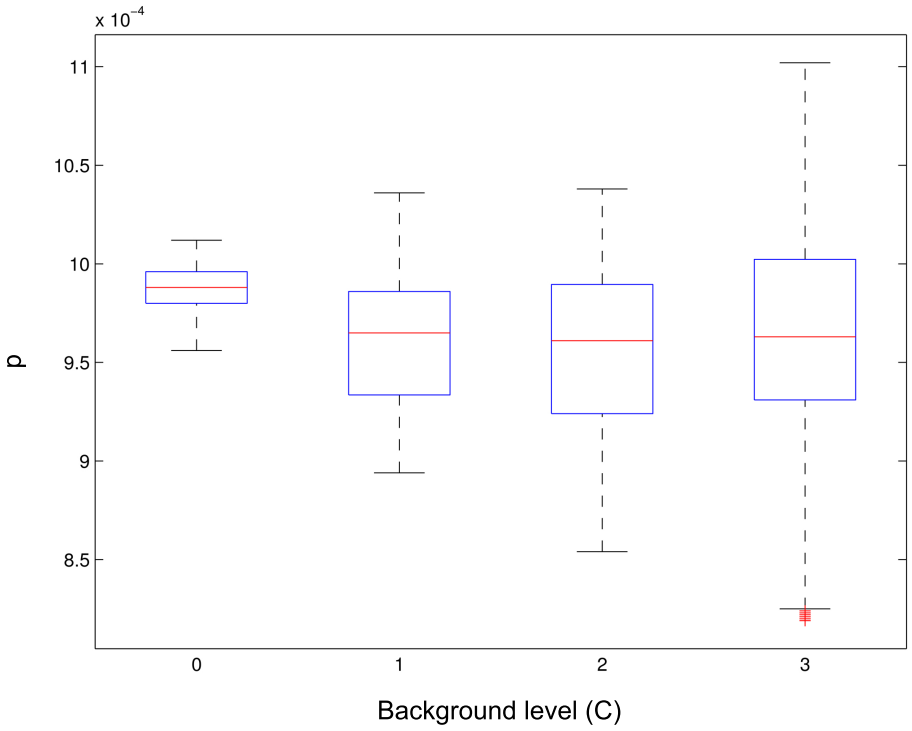


Figure 5.10. Distribution of optimal estimates of p_s (ideal $p_s = 10^{-3}$) for different background levels $C = 0, 1, 2, 3$, sample size $N_s = 10^4$, other parameters: $\mu = 200,\ \sigma = 30,\ a = 2000,\ k = 50$.

5.7.5 Effect of mapping size

The systematic error decreases with the size of the mapping (k) given the same distribution of fragment sizes (Fig. 5.12). The optimal estimate approaches its true value as k approaches the average of the fragment size. As discussed in

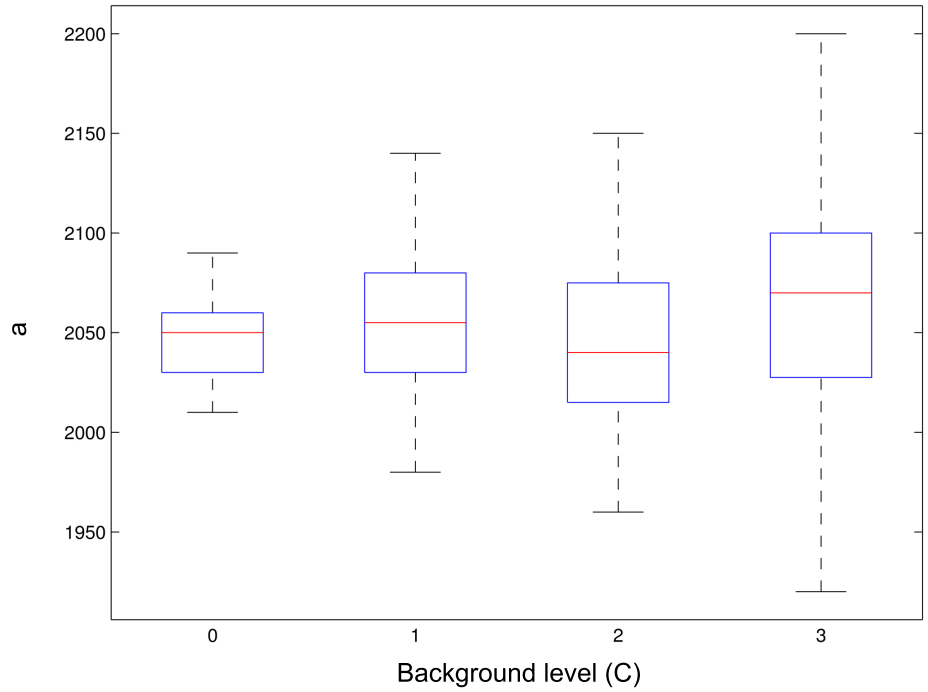


Figure 5.11. Distribution of optimal estimates of a (ideal $a = 2000$) for different background levels $C = 0, 1, 2, 3$, sample size $N_s = 10^4$, other parameters: $\mu = 200,\ \sigma = 30,\ p_s = 10^{-3},\ k = 50$.

Chapter 2, the size of the mapped tag is chosen as a trade-off between increasing the chance of unique mapping and decreasing the chance of a sequencing error present in the tag. Some ChIP-Seq analysis algorithms artificially extend the length of the mapping beyond the length of the tag and up to the mean fragment length (Rozowsky *et al.* (2009b)). My results confirm the benefits of this step for the unbiased parameter estimate from the pileup data.

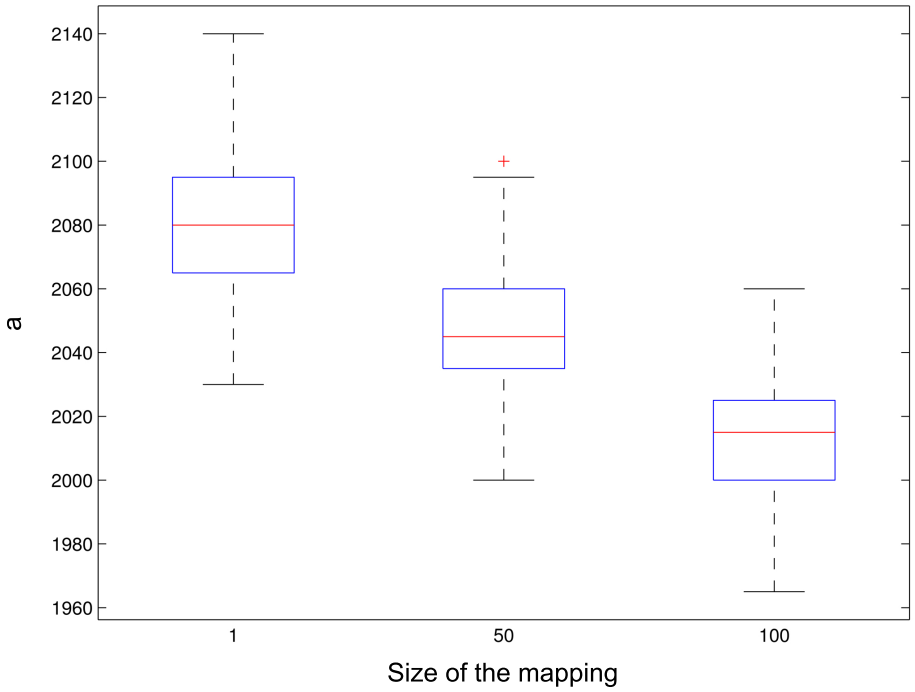


Figure 5.12. Distribution of optimal estimates of a (ideal $a = 2000$) for different mapping sizes $k = 1, 50, 100$, sample size $N_s = 10^4$, other parameters: $\mu = 200,\ \sigma = 30,\ C = 0,\ p_s = 10^{-3}$.

5.7.6 Discarding identical fragments

In the real experiment identical fragment reads (ξ_-, ξ_+) are usually removed to exclude the potential bias introduced by uneven PCR multiplication of the fragments. This step does not pose any problem as long as appearance of identical fragments in the sample is extremely unlikely. Fig. 5.13 shows how the percentage of identical fragments increases with the sample size. When the density of the mapped fragments is sufficiently low the proportion of identical fragments in the sample is negligible. However, when N_s increases the chance of occurrence of identical fragments grows with N_s. Removal of identical fragments in the situation of a large sample, which would be preferable otherwise, unavoidably

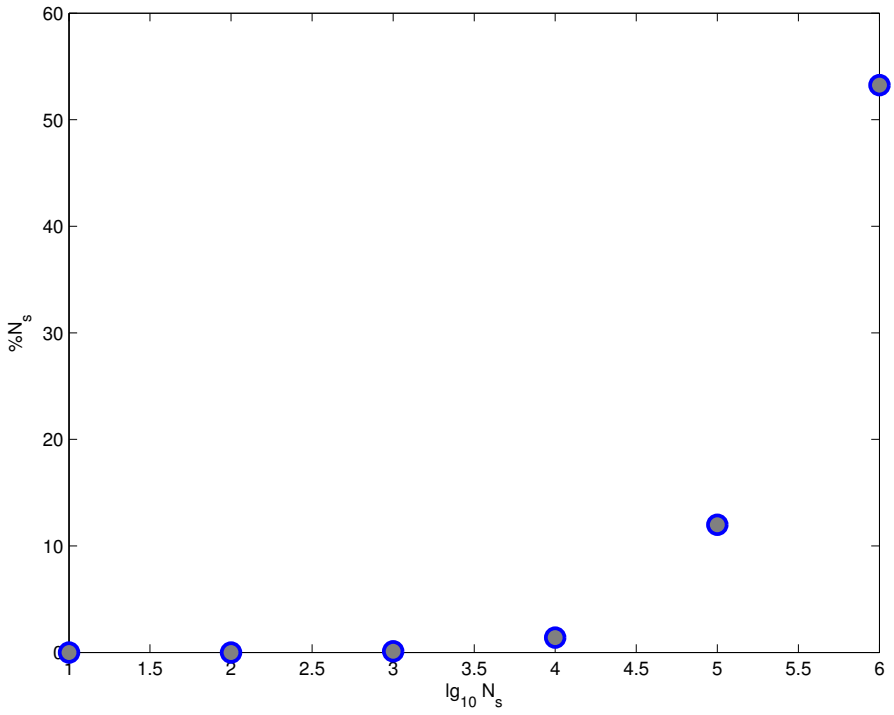


Figure 5.13. The percentage of identical fragments removed from the sample for different sample sizes $N_s = 10^2, 10^3, 10^4, 10^5, 10^6$, other parameters: $C = 0.7,\ a = 2000,\ \mu = 200,\ \sigma = 30,\ k = 50$.

leads to bias in the parameter estimates because the fragments are distributed non-uniformly across the sequence span, so there would be more removals where the density of the fragments is higher (in the peaks of the signal) as compared to the troughs with a low density of fragments. For example, in the case when $N_s \geq 10^6$ on a sequence span of 15 kbp ($X = 15000$), the fragment density being of the order of 60 per base pair, identical fragments constitute a considerable proportion - roughly 12%. Removal of this fraction of fragments would unavoidably bias the parameter estimates. Experimentalists should be cautious when opting for removal of identical reads if the read density exceeds $\sim$ 10 per base pair and perhaps consider other ways of correcting for the sequencing bias. Later in this chapter I will discuss one of these methods.

5.7.7 Effect of data binning

In order to model the fragment map signal it is highly desirable to have smoother data and less noise. As shown earlier (Fig. 5.7) in this chapter reducing the noise in the data helps narrowing the uncertainty of the estimate, where the error was inversely proportional to the square root of the total number of samples N_s. There are three possible methods to smooth the signal

- Data binning, i.e. dividing the truncated region into W-size non-overlapping bins and aggregating the hits that fall within each bin.
- Running average.
- Non-linear regression.

Hereafter, I will compare alternative methods of smoothing the data focusing on data binning and running average.

When binning the hit counts in pileup data it is important to keep in mind that larger bin sizes introduce bias into the parameter estimation, which is especially noticeable when it becomes comparable to the scale of the data (Fig. 5.14). This result is expected since binning the data not only decreases the noise but also negatively impacts the resolution of the data, which is essential to estimate the parameters with precision. Therefore, it is important to avoid binning if possible or choose the bin size to be significantly smaller than the scale of the data ($W \ll L$).

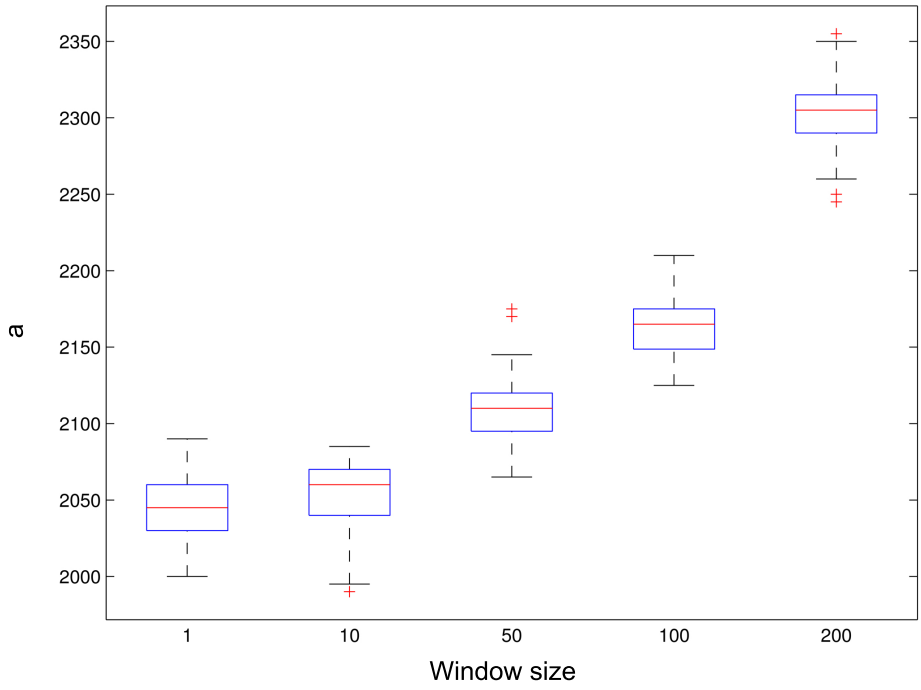


Figure 5.14. Distribution of optimal estimates of a (ideal is $a = 2000$) for different bin sizes $W = 1, 10, 50, 100, 200$, other parameters: $N_s = 10^4,\ C = 0,\ \mu = 200,\ \sigma = 30,\ k = 50,\ p_s = 10^{-3}$.

5.8 Running average as an alternative to binning

If one still chooses to reduce the noise, instead of binning the data it would be a better choice to use a running average algorithm. Smoothing the data with a running average will reduce the sequence bias but will not hugely affect the parameter estimates if the window size is chosen not too large. On Fig. 5.15 one can see that with a window size smaller than $W = 200$, the systematic error remains approximately the same as without smoothing at all. It is only when the window increases beyond $W = 200$ that it would possible to see significant

deviation from the initial estimate. This is easy to understand as the scale of the model is $L \sim 2000$, so the window should be considerably smaller than this characteristic size of the data and $W = 200 \ll 2000$ still fulfils this requirement. In conclusion, the size of the window W must be chosen as a trade-off between

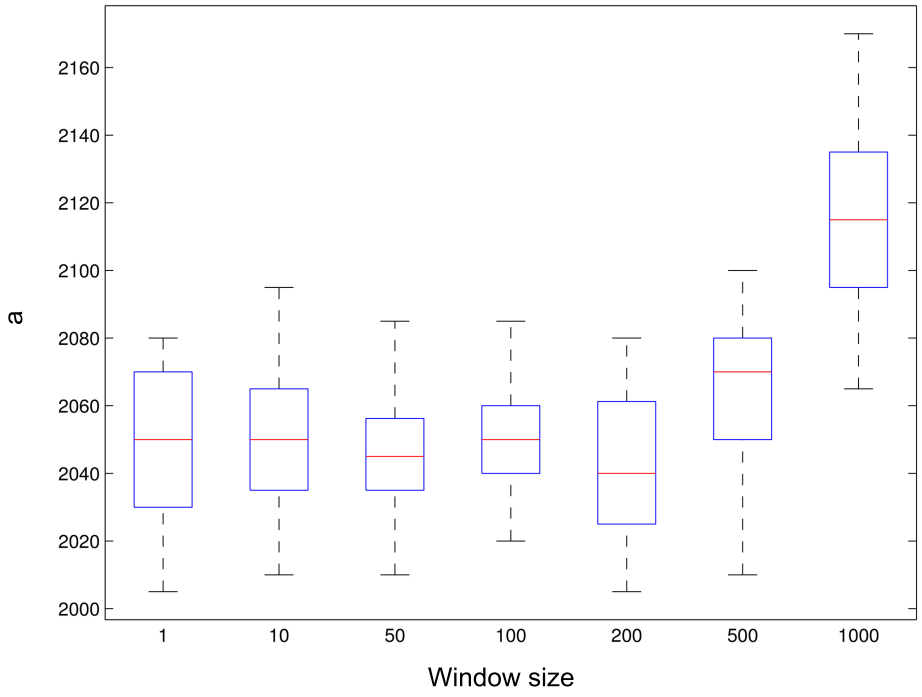


Figure 5.15. Distribution of optimal estimates of a (ideal is $a = 2000$) for different window sizes $W = 1, 10, 50, 100, 200, 500, 1000$, other parameters: $N_s = 10^4,\ C = 0,\ \mu = 200,\ \sigma = 30,\ k = 50,\ p_s = 10^{-3}$.

data robustness and resolution.

5.9 Sequencing bias

As demonstrated in Chapter 2, fragments that have a higher percentage of GC are amplified during the experiment to a higher extent than others resulting in sequence dependent bias, which hampers fair data analysis. The average content of the whole sequence is roughly 50% GC and 50% AT, however on a lower scale the sequence composition is highly uneven, i.e. there exist islands of high density of %GC contributing to inflated hit counts in those regions as compared to the regions with a low density of GC where the hit count is below average. We need to determine the size of the window for which the GC content is less uneven on average. Smoothing the data using this window would help reduce the discrepancies associated with the local composition. So, a window size larger or equal to this chosen window should be used to smooth the data in order to average out the noise associated with GC bias. As seen on Fig. 5.16, a larger

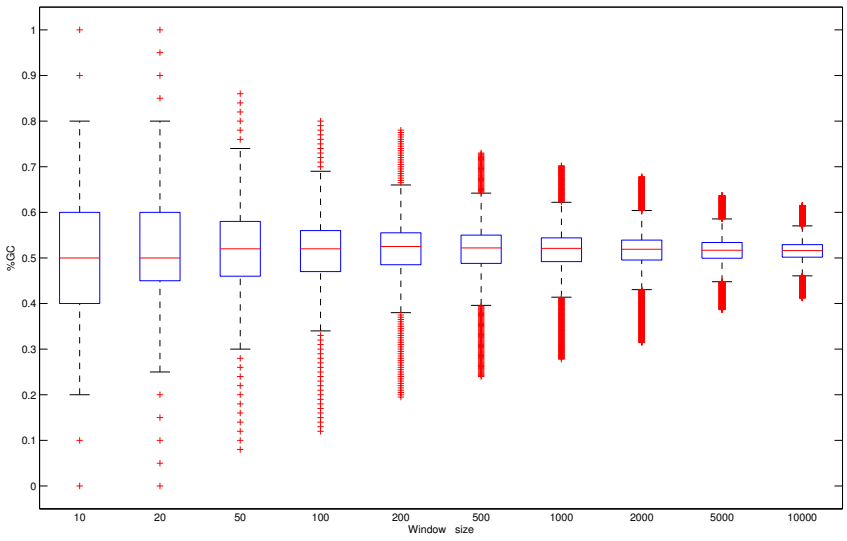


Figure 5.16. Distribution of GC content as a function of the window size.

smoothing window size narrows the distribution of the GC content around 50%.

While a window size of $W = 20$ sees the spread in GC content between 0.2 and 0.8, when the window size is as large as $W = 1000$ the diapason of GC is already within $0.4 < \%GC < 0.6$, reducing the variability by a factor of 3. It appears that analysis of the data with a characteristic size of $L \gg W_c \approx 1000$ will suffer less from sequence bias than the data with a smaller characteristic size.

In summary, the choice of a smoothing window size must be a trade-off between the systematic error introduced by signal averaging (Section 5.7.7) and the negative impact of GC bias, because a larger window size allows to minimise sequence bias.

$$W_c < W \ll L$$

Chapter 6

Case study

6.1 Summary

In this chapter [1], I build a stochastic model of RecBCD's action on a DNA double strand break (DSB). The goal of this work is to use this model to estimate the key biophysical parameters describing the mode of action of RecBCD *in vivo*. Given a set of parameters, the model can be used to assign a likelihood to the experimental data. Whichever set of parameters maximises this likelihood will provide the estimate we seek.

The model is of a hybrid "mechanistico-genomic" nature. On one hand, it draws from traditional stochastic modeling (discrete-time Markov chains) to represent the progression of the RecBCD complex on DNA and the various stages of DNA resection after a DSB; on the other hand, it incorporates precise genomic information to fix the position of Chi sites which are the master triggers of this

[1] Most of this chapter has been written by Vincent Danos and Meriem El Karoui based on my mathematical model of DSB mediated ChIP-Seq data. A large part of it is contained in the Supplementary material for the paper by Cockram *et al.* (2015).

process. It is this somewhat unusual combination of mechanistic and genomic information which allows us to exploit the data quantitatively and use it to investigate some of its underpinning biophysics.

This chapter is organised as follows. First, I review the existent knowledge and detail the simplifications we make to obtain the structure of our model. As is generally the case, the exercise of setting up the model is an excellent way to integrate the current biological understanding of the process. With the basic modeling choices in place, I analyse the mathematical structure of the model. I find that the model is simple enough that one can derive a closed formula for the resected single-stranded DNA segments produced by the idealised stochastic process. Then, I use this formula to compute the likelihood of the actual data according to various choices of parameters, and narrow down on a most likely set thereof. Finally, I present the discussion of the results.

6.2 Model

6.2.1 The mechanism of action of RecBCD

Extensive biochemical characterisations reviewed in Dillingham & Kowalczykowski (2008) and in Smith (2012) demonstrate that the RecBCD complex loads on a DSB and translocates along DNA until it recognises a Chi site. Chi recognition is not certain, and RecBCD may read through several Chi sites before recognising one. Before recognition, the RecB and RecD motors are both engaged. As RecB is slower than RecD, a single strand loop accumulates ahead of RecB. Upon recognition, RecB becomes the lead motor and RecBCD's activity is modified so that the 5' strand is degraded, while RecA gets loaded on the 3'

overhang. The loop formed prior to Chi recognition contributes to the 3' resected end that starts at the recognised Chi. Fig. 6.2 summarises the two stages of the resection process. Eventually RecBCD stops loading RecA and dissociates from DNA. This model is equally compatible with biochemical data of RecBCD activities obtained when the concentration of magnesium exceeds that of ATP or when the concentration of ATP exceeds that of magnesium.

6.2.2 Modeling choices

We translate this molecular knowledge in a series of modeling decisions and simplifying assumptions which we detail below. First, we model the recognition of a Chi site as a *stochastic* event. This seems natural as it is well observed that Chi recognition is not deterministic, and indeed only a stochastic model will allow us to get quantitative estimates on this important aspect of the process. Specifically, we assume that Chi sites are recognised by RecC with a probability p_χ which does not depend on the distance from the DSB, nor does it depend on the number of Chi sites previously encountered by RecBCD.

We also model RecBCD's translocation in a stochastic way. The specific translocation mode depends on whether a Chi site has already been recognised or not. *Before* recognition, to take into account the different speeds v_B, and v_D of RecB and RecD, we distinguish two types of steps:
- one where the RecB and RecD motors move in unison with probability p_-;
- one where only the faster one, RecD, moves with complement probability $p_+ = 1 - p_-$.

In this mode, the mean ratio of the distances covered by RecD and RecB after any number of steps is given by $1/p_-$ (see §6.2.4), hence the speed ratio of the

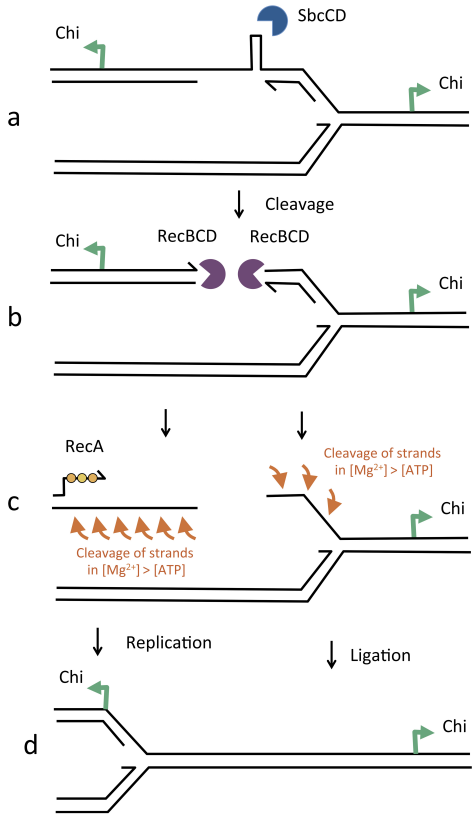


Figure 6.1. Hypothetical mechanism for the conversion of a two-ended break to a one-ended break. (A) SbcCD enzyme cleaves a hairpin formed on the lagging strand at the site of an interrupted palindrome. (B) The two ends are processed by RecBCD enzyme. (C) The origin-proximal end is processed to a Chi site and RecA protein is loaded. The origin-distal end is processed up to the replication fork avoiding recognition of an origin-distal Chi site. (D) The origin-proximal end recombines with the sister chromosome and the nick left on the origin-distal side is ligated.

two motors is given by $v_B/v_D = p_- \leq 1$. This means that, consistently with Taylor & Smith (2003), the model assumes that $v_B \leq v_D$.

Together with p_χ, estimating the speed ratio p_- is a key objective of the model.

After Chi recognition, the 3' strand is extended further and RecA is loaded on

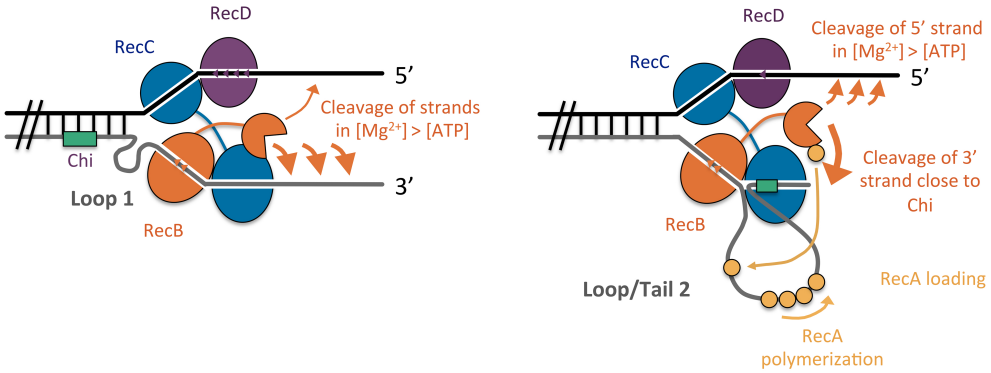


Figure 6.2. Sketch of DNA resection by RecBCD. Left panel - before Chi recognition: the RecB and RecD motors move along DNA and the RecB motor lags behind the RecD one; a loop forms ahead of RecB. Right panel - after Chi recognition: the entire RecBCD complex undergoes a conformational change which directs RecB's nuclease activity to the 5' strand, and induces the loading of RecA on the 3' one. In this schematic representation, the Chi site is shown held in its recognition site. However, the Chi site will be released either by disassembly of the RecBCD complex or at some point prior to this and the second single-stranded region will be converted from a loop to a tail.

this strand. In this second mode, we assume that RecA is loaded uniformly on the single strand and we suppose that there is a constant probability p_{stop} for RecBCD to stop loading RecA (or to fall off) at each step.

We write τ_1 for the length of the single stranded loop (on the 3' strand ahead of RecB) at the time a Chi site is recognised. The mean value of τ_1 depends linearly on the distance of the said Chi site from the original DSB - the further the Chi site, the longer the loop. Similarly, we write τ_2 for the length of the single strand extension after Chi recognition and until RecBCD stops loading

RecA (and possibly dissociates from DNA). Differently from τ_1, the value of τ_2 does not depend on which Chi site is recognised.

We also assume that the RecB subunit starts loading RecA only after Chi recognition (on the 3' resected strand). This means that the resected segment will begin at whichever Chi site is recognised and will have a total length of $\tau_1+\tau_2$. And finally, we assume that whenever RecBCD falls off DNA *before* having recognised a Chi, the obtained single strand is not observable in the experiment as no RecA has been loaded.

Putting our choices together, we obtain a stochastic model (a discrete-time Markov chain) which generates the 3' resected segment onto which RecA is loaded. The model uses a restricted set of parameters $\mathcal{P}$ which consists of p_χ, p_-, and p_{stop}. Its overall structure is described in Fig. 6.3.

The model also includes the spatial configuration of Chi sites on the DNA. Let $I = \{1, \ldots, c\}$ be the set indexing the Chi sites in order of appearance after the DSB, we write λ_i for the distance of the i^{th} Chi site from the DSB with $\lambda_1 < \ldots < \lambda_c$. Because we know the genome sequence of the strain of interest, and the sequence of the Chi sites (5'-GCTGGTGG-3'), there is no need to make these sites explicit parameters of the model. It has been suggested that other sites can act as Chi-like motifs Cheng & Smith (1987), but these are weaker and we do not take them into account.

6.2.3 Variants

There are several other modeling options we could have considered. Let us mention two. A natural way to enrich the model would be to allow for reversible translocation of the motors, following the lines of the toy bimotor

model developed in Ref. Stukalin *et al.* (2005b). This would result in a smoother behaviour and potentially describe better the finer details of the biophysics of the motors. Another natural elaboration is to assume stochasticity in the parameters v_B, v_D governing the speed of the motors on DNA. Indeed, it has been shown recently that, *in vitro*, the pre-recognition translocation speed of RecBCD is itself fixed for an entire run by initial stochastic molecular events Liu *et al.* (2013b). We discuss later whether incorporating this particular observation could result in a useful refinement of our model. With the simple model which we employ first, there is no need to predict the kinetics of the operation of RecBCD, and therefore no need to calibrate the time unit implicitly in our discrete-time modeling.

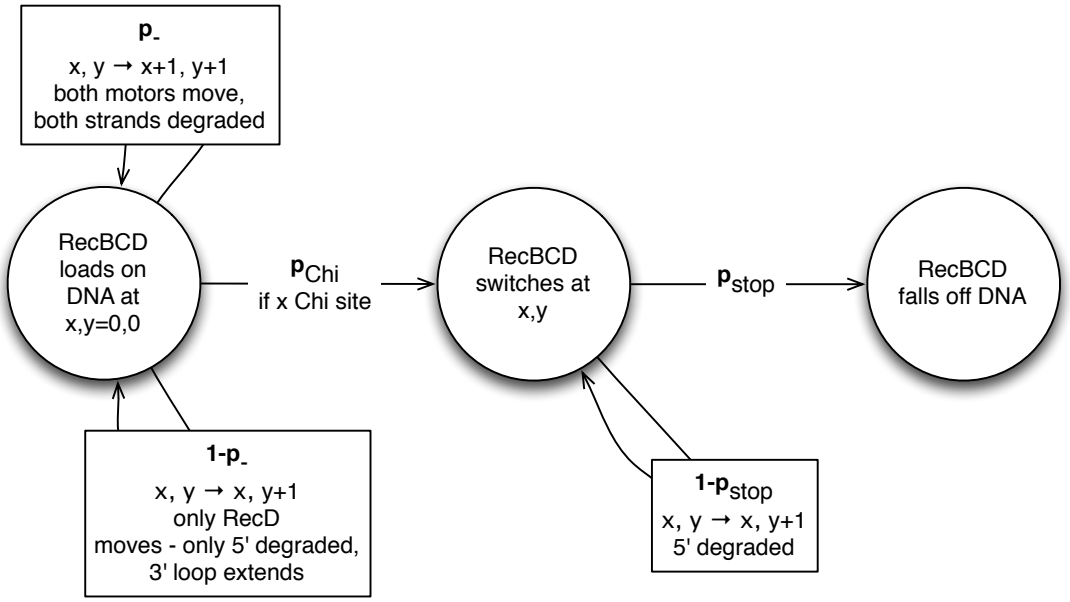


Figure 6.3. Decision tree for the model of DSB resection by RecBCD: x, y represent the respective DNA positions currently read by RecB and RecD; when x is a Chi site, with probability p_χ RecBCD switches to the mode where only 5' is degraded, else both motors continue to translocate along the dsDNA as before.

6.2.4 Derivation of the single strand distribution

There are three sources of randomness which jointly determine the segment produced by RecBCD:
- Y the (index of the) Chi site recognised by RecC,
- τ_1 the length of the single strand loop at the time a Chi site is recognised, and
- τ_2 the additional distance travelled by RecBCD after having recognised a Chi site.

In the following, we derive a simple formula for the distribution of these segments and their total length $\tau = \tau_1 + \tau_2$, and for the probability $Pr(x|\mathcal{P})$ that a nucleotide x is part of a segment.

Our first step is to calculate τ_1, assuming the Chi site recognised is at distance λ from the DSB. The value of τ_1 is given by the number of steps where RecB has not moved, and which have therefore resulted in extending the loop ahead of RecB, by the time RecB reaches λ.

Let $\mathcal{B}^-(X = k|n, p) = \binom{n+k-1}{k} p^n (1-p)^k$ denote the probability that a random variable X, distributed according to a *Negative Binomial* with parameters $n > 0$ and $p > 0$, takes a non-negative integer value k. The values of X track the number of failures needed to obtain n successes, each trial being independent, and p being the common probability of success. This translates directly to our setting, with n being the number of moves of RecB prior to Chi recognition, and p being p_- the probability of RecB moving.

Hence, when RecB arrives at position λ, RecD is ahead at position $\lambda + \tau_1$, with the distance between the two, namely τ_1, being distributed as:

$$Pr(\tau_1 = k) = \mathcal{B}^-(\tau_1 = k|\lambda, p_-) \tag{6.1}$$

From this, we can write an explicit formula for the mean length of the loop as a function of λ - the position where recognition happens (measured as a distance from the DSB):

$$\tau_1 = \lambda(1 - p_-)/p_- \tag{6.2}$$

This formula is useful to evaluate the impact of p_- on the length of the loop. We can see from this that the mean ratio of the distances covered by the two motors is the mean of $(\lambda + \tau_1)/\lambda$. As a negative binomial has mean $n(1-p)/p$, we find that the mean speed ratio is $1 + (1 - p_-)/p_- = 1/p_- = v_D/v_B$. In other words, p_- is none other than the v_B/v_D speed ratio.

Our second step is to evaluate the additional distance τ_2 travelled by RecBCD (with only RecB engaged, and the 5' strand being degraded) after recognition of the Chi site and until RecA loading stops. The 3' strand is extended until RecBCD stops loading RecA and/or dissociates, hence τ_2 follows a geometric distribution with parameter p_{stop}. We will write $\mathcal{G}(X = k|p) = \mathcal{B}^-(X = k|1, p) = (1-p)^k p$ for the geometric distribution of parameter p where the random variable X tracks the number $k \geq 0$ of failures.

Taking into account the fact that τ_1 and τ_2 are independent variables, we get the following expression for the distribution of the total length $\tau = \tau_1 + \tau_2$ of the segment produced by RecBCD, conditioned on the Chi site at λ being recognised:

$$Pr(\tau = z|\lambda) = \sum_{k=0}^{z} \mathcal{B}^-(\tau_1 = k|\lambda, p_-)\mathcal{G}(\tau_2 = z - k|p_{stop}) \tag{6.3}$$

The next step is to obtain the joint distribution of the 2D-random variable (τ, λ) where τ is the length of the segment, and λ is the distance from the DSB where the segment starts. As in our simple model, the DNA is always degraded up to the recognised Chi, λ takes values in the set of distances of Chi sites from the DSB, namely $(\lambda_i; \times i \in I)$. The index Y of the Chi site eventually recognised is

distributed as $\mathcal{G}(Y = i + 1|p_\chi)$ for $0 \leq i < |I|$. (The offset by 1 comes from the fact that we start numbering Chi sites at 1).

Putting our calculations together we get the joint distribution:

$$Pr(\tau = z, \lambda = \lambda_i) = \mathcal{G}(Y = i + 1|p_\chi) \sum_{k=0}^{z} \mathcal{B}^-(\tau_1 = k|\lambda_i, p_-)\mathcal{G}(\tau_2 = z - k|p_{stop}) \tag{6.4}$$

From this one can compute the *hit probability* of a nucleotide x, that is to say the probability of x to be included in the segment defined by τ and λ:

$$Pr(x|\mathcal{P}) = \sum_{\{i \in I | \lambda_i \leq x\}} \sum_{\{z \geq x - \lambda_i\}} Pr(\tau = z, \lambda = \lambda_i) \tag{6.5}$$

Note that the hit probability at x is zero unless one of the Chi sites before x is recognised. In particular, a 'runaway' RecBCD which fails to recognise *any* Chi, generates no segment and induces no RecA loading. Note also that the sum $\sum_x Pr(x|\mathcal{P})$ is not 1, as many x's receive hits simultaneously. In fact, $\sum_x Pr(x|\mathcal{P})$ is the mean number of hits, that is to say the mean length of the resected segment.

The hit probability depends strongly on the particular set of parameters $\mathcal{P}$ and we will exploit this dependency to estimate our three parameters: p_χ, p_-, and p_{stop}. By sampling the set of parameters, we can compute for each set how likely the data are according to this set -a quantity defined as the *likelihood* of the parameter set (see below for a precise definition). Provided we can do this sampling efficiently, we can obtain a precise 'heat map' of the parameter space, whose peaks will denote the maximally likely values of the parameters.

An approximation

In order to sample efficiently our parameter space, we use an approximation of $Pr(x|\mathcal{P})$ and replace τ_1 by its mean $\lambda_i(1-p_-)/p_-$. This is equivalent to supposing that the speed ratio v_B/v_D is constant.

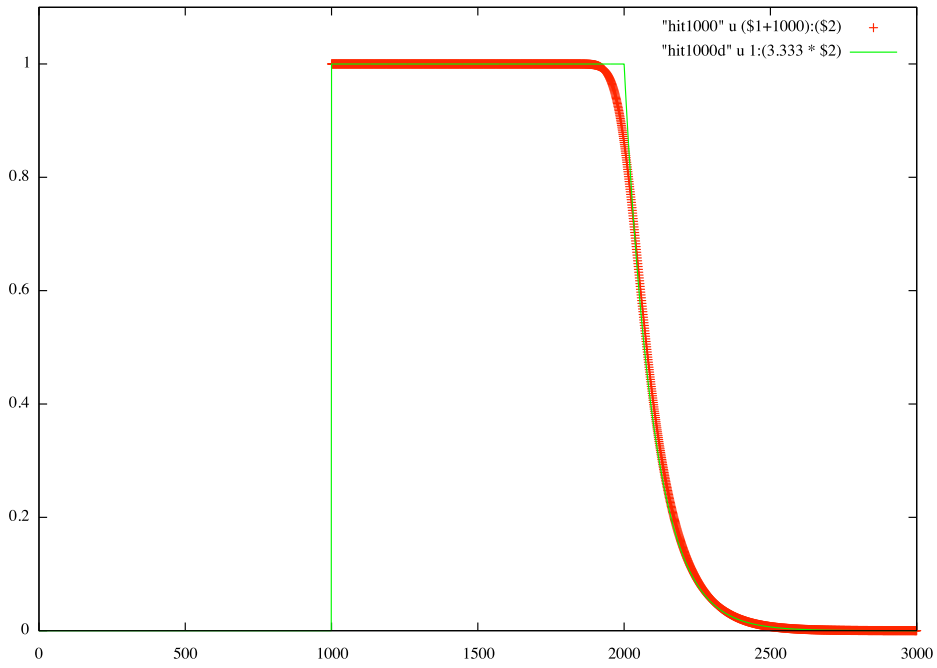


Figure 6.4. We use here for comparison a single transition site positioned at 1000 with $p_{stop} = 0.01$, $p_- = 0.5$, $p_\chi = 0.3$. Hence $\alpha = 2$ and the approximation (green) of $Pr(x|\lambda = 1000, \mathcal{P})$ is flat until position 2000. We see that it is quite close to the exact calculation (red).

With this approximation the expression for the hit probability, conditioned on recognition happening at λ_i, simplifies to:

$$Pr(x|\lambda = \lambda_i, \mathcal{P}) = \mathbf{1}_{\{\lambda_i \leq x \leq \alpha\lambda_i\}} + (1 - p_{stop})^{x-\alpha\lambda_i}\mathbf{1}_{\{x>\alpha\lambda_i\}} \tag{6.6}$$

with $\alpha = 1+(1-p_-)/p_-$, and $\mathbf{1}_A$ the indicator function for A. This approximation

incurs a negligible loss of precision as we see in Fig. 6.4 for a set of representative parameters. In general, the normalised error on the hit probability will be of the order of the coefficient of variation $1/\sqrt{\lambda_i(1-p_-)}$ which quickly becomes negligible as λ_i increases, as the closest Chi sites stand at $3kb$ from the DSB.

6.3 Data

The data consist of six data sets corresponding to the strains carrying 1 to 6 Chi sites at $3kb$ on the origin proximal side of the DSB (derived from Cockram *et al.* (2015)). We focus on a $100kb$ region X on the origin proximal side of the DSB, as beyond this distance the signal reaches background noise level. This region does not contain any DSB-independent RecA binding loci thus allowing us to apply the model described above on the entire region.

We use as input the $50bp$ reads mapped on the reference genome using novoalign version 2.0 (Hercus (2012)). In order to compensate for any bias introduced by PCR amplification of DNA fragments before sequencing, multiple duplicate fragments (fragments starting and ending at the same positions) are replaced by a single 50bp read. The data are then processed by dividing the region in $250bp$ long non-overlapping bins and aggregating the reads that fall within each bin. The size of the bin is chosen as a trade-off between data robustness and resolution. As the bin size is much smaller than the expected size of a single strand coated by RecA (which is in the order of several kb) resolution should be minimally affected.

It remains to define and measure the *background* level of the RecA signal. To do this, we assume that there is no RecBCD-mediated loading of RecA before the Chi sites which stand closest to the break at about $3kb$. The RecA signal seen

in this Chi-less region is treated as background, and we subtract its average level from the the binned data before comparison with the model.

6.3.1 Comparing model and data

In order to compare our model and the data, we rank sets of parameters $\mathcal{P}$ according to the probability they assign to the processed data within the region of observation X (see above). For adequate comparison the model results are aggregated in bins of $250bp$. One of the parameters which has a dimension, namely p_{stop} which is the inverse of a distance, is divided by the bin size 250. The probability of detecting a nucleotide x in X (or the probability of observing a hit at x) can be written as:

$$F(x|\mathcal{P}) = \frac{Pr(x|\mathcal{P})}{\sum_{x \in X} Pr(x|\mathcal{P})} \tag{6.7}$$

This simple model assumes that the DNA fragments that are read are of the same length and located at identical positions as the initial single strand fragments onto which RecA is loaded. It also assumes that the DNA fragments are distributed uniformly and not biased by the sequence. This assumption is supported by the following arguments: (i) DNA fragments produced by sonication at the start of the ChIP process are unaffected by RecA binding. Hence fragments whether covered by RecA or not have the same probability of being sheared. (ii) As said above, PCR generated duplicates (identical fragments) are discarded, and then only the first $50bp$ (out of an average length of the fragment of $200bp$) of each remaining sequenced fragment is retained in the final hit count. (iii) The pileup data generated from the input samples (without RecA pulldown) show no sequence bias in the double strand break region (Fig. 6.5) (we note that the sequence GC content does not vary significantly in this region which does

not contain horizontally transferred segments). (iv) While *E. coli* replication mechanism will lead to regions close to the origin showing a higher DNA copy number than regions close to the terminus of replication, the variability on the analysed region given a cell doubling time of 40 minutes is 0.25% and is therefore negligible.

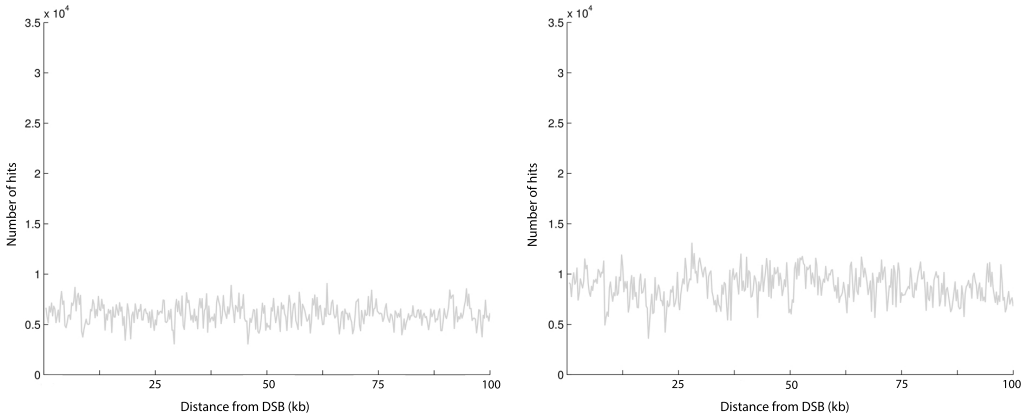


Figure 6.5. Two replicates of the hit counts per 1 kbp obtained from sequencing without RecA immuno-precipitation on the region of interest

6.3.2 Parameter estimation

The amount of DNA obtained before PCR amplification (of which the only role is to produce enough material for sequencing) is small enough that with a very high probability, no two reads come from the same individual DSB event. This means that the hits recorded from each read are approximately statistically independent. Hence we can conceptualize the experiment as drawing repeatedly and independently from a pool of nucleotides (the total amount of DNA collected in a given experiment), some of them being marked (included in a resection

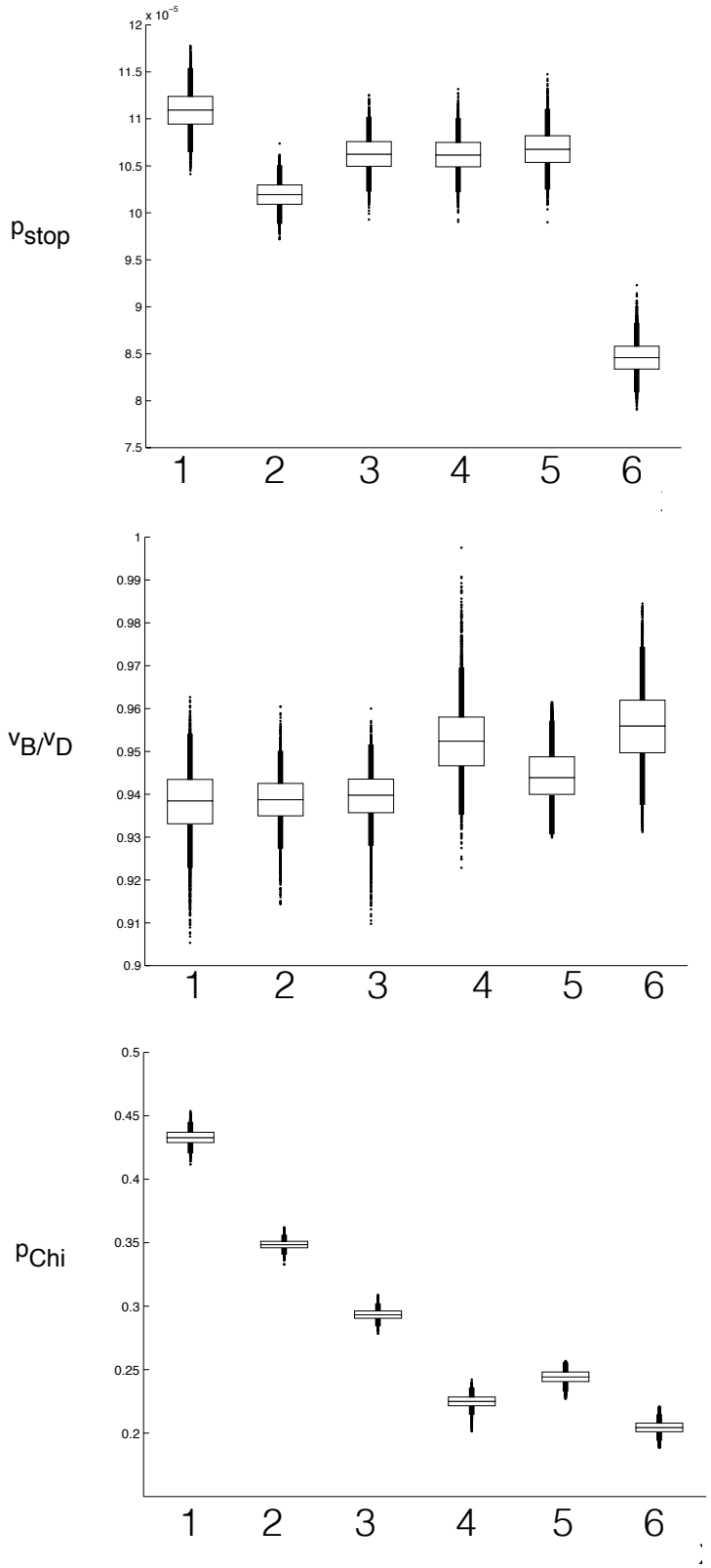


Figure 6.6. Boxplots describing the probability distribution of the three model parameters p_{stop} (top), $p_- = v_B/v_D$ (middle), and p_χ (bottom) for each of the six strains (enumerated from 1 to 6 on the x-axis). The parameter distributions are obtained by a Metropolis-hastings algorithm (see Section 2.6.4)

segment), and some not. Then again, because the sample taken from the pool is extremely 'thin', we can assume that the drawing is with replacement, and follows therefore a multinomial distribution. The likelihood of the data, can then be written to a very good approximation in this simple way:

$$\mathcal{L}(\mathbf{n}|\mathcal{P}) = C(\mathbf{n}) \prod_{x \in X} F(x|\mathcal{P})^{n_x} \tag{6.8}$$

where X is our $100kb$ region of interest, x ranges in the observation region X, $\mathbf{n}$ is the sequence n_x of x's hit counts in the data, and $C(\mathbf{n})$ is a multinomial coefficient. Taking a logarithm of the above, and forgetting $C(\mathbf{n})$ which does not depend on $\mathcal{P}$, and therefore plays no role in the maximisation of $\mathcal{L}$, we arrive at the following objective:

$$\sum_{x \in X} n_x \log F(x|\mathcal{P}) \tag{6.9}$$

that is to say we wish to find the value of $\mathcal{P}$ which maximises the above expression.

To estimate this best set of parameters, we follow a simple strategy and sample the $[0, 1]^3$ interval as follows:
- $[0.5, 1]$ with step size 10^{-2} for $p_- = v_B/v_D$,
- $[0.1, 0.7]$ with step size 10^{-2} for p_χ,
- $[0.810^{-4}, 1.2 \times 10^{-4}]$ with step size 4×10^{-6} for p_{stop}. The sampling was implemented using a Matlab script (available upon request) to compute the log-likelihood and locate the global maximum. The obtained optimal values are shown in Fig. 6.6. The box plots describe the likelihood of each parameter in the neighbourhood of these optimal values (see below). Also, MLE estimates of the parameters are summaries on Table 6.8.

6.3.3 Discussion

As one can see in Fig. 6.7, our parsimonious model reiterates the data rather well for the optimal parameters. This suggests that the model has indeed captured some of the salient aspects of the mechanisms at play in the real system. To gauge the local log-likelihood distribution at higher resolution than our initial grid sampling, we ran a Metropolis-Hastings algorithm starting at the previously identified global maximum. The jump sizes are taken to be uniformly distributed within $\pm\epsilon$, where ϵ is the resolution of the mesh used in the grid sampling (see right above). The associated random walk samples the immediate neighbourhood of our best estimate (10000 steps for each data set). The results shown as box plots in Fig. 6.6 confirm the presence of strong local maxima.

With the obtained parameters, the size of the loop, namely τ_1 in our notations, will be of the order of $0.05/0.95 \times 10^2 kb \sim 5kb$ at the far $100kb$ end of the Chi site range in X (the Chi sites most distant from the DSB), while τ_2, the other component of the length of the single strand is independent of the site of recognition and of the order of $1/p_{stop} \sim 10kb$. The resected segment will take a range of values which is bounded below by τ_1. As the efficiency of the search for an homologous sequence for repair depends on the length of the segment, the τ_1-"loop" might have a determinant role to play. In addition, the loop also plays a role in the loading of RecA, and therefore efficient loading might also depend on τ_1.

The values predicted for p_{stop} and $v_B/v_D = p_-$ are stable across the 6 different data sets, as they should, as these values are meant to capture mechanistic parameters that are independent of the conditions of the experiments. On the other hand, the value of the recognition probability varies from one data set to the next: there is a decrease in the predicted p_χ as the number of Chi sites in the

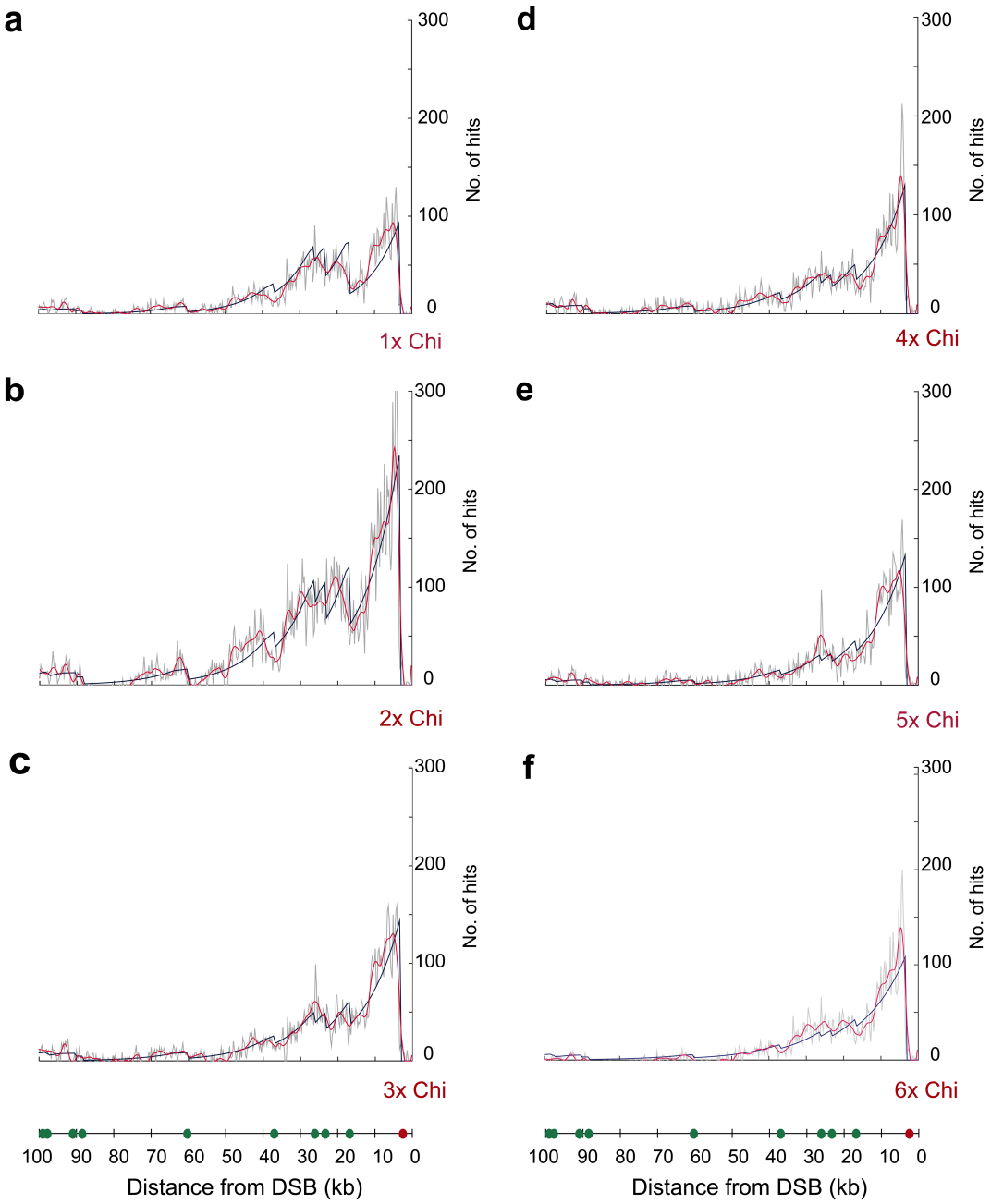


Figure 6.7. (a-f) 1-6 Chi sites. Blue line - prediction of the model. Red line-smoothed data (Loess filter with bandwidth 5700 nucleotides, span 0.057). Grey line - the number of hits per 250 bp window normalized to the total number of reads. Green circles - endogenous Chi site, Red circles -Chi arrays.

No. of Chi sites in array	1	2	3	4	5	6
p_χ	0.43	0.35	0.29	0.23	0.24	0.20
V_B/V_D	0.94	0.94	0.94	0.95	0.95	0.96
Ps	1. 1x 10^{-4}	1 x 10^{-4}	1 x 10^{-4}	1 x 10^{-4}	1 x 10^{-4}	0.84 x 10^{-4}

Figure 6.8. The three parameters of the model $\mathcal{P}$ were inferred independently on each dataset using a MLE

initial array increases. The Chi sites in the array are separated by only $10bp$. The trend which we observe in p_χ is likely due to them being placed too close in the array for the Chi recognition subunit, RecC, to work independently on each site. This means in turn that the most robust estimate of p_χ is likely to be found in the case of the array containing 1 Chi site. We will focus on this data set below.

6.3.4 Model comparison

Estimates of v_B/v_D using our initial model (referred to below as the basic model) are close to 1 and substantially higher than reported in the literature (Smith, 2012). This raises the question as to whether the signal in the data is strong enough to allow a correct estimation of v_B/v_D or whether the model would fit the data equally well if v_B/v_D was simply fixed to 1. In that case, the actual v_B/v_D may still be different from 1 as observed *in vitro*, but this would suggest that the data do not allow its correct estimation. To compare the performance of the basic model when fixing v_B/v_D to 1 or estimating it from the data, we used a BIC (Bayesian Information Criterion (Kass & Raftery, 1995; Schwarz *et al.*, 1978)) score. BIC takes into account the log-likelihood (L) of the data but penalises models with higher complexity (i. e. a larger number of independent

parameters (q)) to a greater extent than conventional log-likelihood ratio tests. The BIC score is defined as follows:

$$BIC = -2L + 2q \log n \tag{6.10}$$

where n stands for the total number of observations $n = \sum_x n_x$. Note that we use the objective function defined in Eq. 6.9 instead of the true log-likelihood L to calculate the BIC score in Eq. 6.10. The objective function differs from the true log-likelihood by a function dependent on the data only ($\log C(\mathbf{n})$ in Eq. 6.8) but not on the parameters of the model. This is a convenient strategy since the part of the log-likelihood function independent of the parameters is not necessary to compare the models. Table 1 shows the BIC scores of the models where v_B/v_D is either fixed to 1 or estimated. In all cases except the data set with an array of 6 Chi sites, the model where v_B/v_D is estimated from the data is strongly preferred, indicating that v_B/v_D is important to explain the data. In the case of the 6 Chi array, most of the signal is concentrated at the array and there is little signal away from the DSB. The estimation of v_B/v_D is directly dependent on τ_1, and τ_1 is linearly dependent on the distance from the DSB and is better estimated if there is enough signal away from the DSB. It is therefore not surprising that in that dataset v_B/v_D cannot be estimated reliably.

6.3.5 Mixture model

So far, our model is assuming a constant immutable ratio between the velocities of the two motors in RecBCD. But, recent *in vitro* experiments Liu *et al.* (2013b) demonstrate that, in fact, RecBCD operates (at the single molecule level) with a bimodal distribution of velocities. It is tempting to investigate whether a mixture

Table 6.1. BIC scores computed for both models under consideration and all 6 strains with different number of Chi sites

.

N Chi	BIC($v_B/v_D = 1$)	BIC($v_B/v_D < 1$)	Best Model
1	92145	92087	v_B/v_D estimated (very strong)
2	188849	188701	v_B/v_D estimated (very strong)
3	111294	111197	v_B/v_D estimated (very strong)
4	90378	90319	v_B/v_D estimated (very strong)
5	80722	80652	v_B/v_D estimated (very strong)
6	86471	86481	v_B/v_D fixed (strong)

between two modes described by different sets of parameters would explain the data better. The new model can be described as follows:

$$Pr'(x \mid r, \mathcal{P}_1, \mathcal{P}_2) = r \cdot Pr(x \mid \mathcal{P}_1) + (1 - r) \cdot Pr(x \mid \mathcal{P}_2) \tag{6.11}$$

where r is the probability of choosing the first set of parameters $\mathcal{P}_1$ and $1 - r$ is the probability of choosing $\mathcal{P}_2$ accordingly.

To estimate this best set of parameters, we follow a simple strategy and sample the $[0, 1]^6$ interval in two rounds as follows:

Step 1:

- $[0, 1]$ with step size 10^{-1} for $p^1_- = v^1_B/v^1_D$,
- $[0, 1]$ with step size 10^{-1} for p^1_χ,
- $[0, 1]$ with step size 10^{-1} for $p^2_- = v^2_B/v^2_D$,
- $[0, 1]$ with step size 10^{-1} for p^2_χ,
- $[0.8 \times 10^{-4}, 1.44 \times 10^{-4}]$ with step size 4×10^{-6} for p_{stop}.
- $[0.5, 1]$ with step size 10^{-1} for r

Step 2:

- $[0.8, 1]$ with step size 2×10^{-2} for $p^1_- = v^1_B/v^1_D$,
- $[0.2, 0.4]$ with step size 2×10^{-2} for p^1_χ,
- $[0.4, 0.6]$ with step size 2×10^{-2} for $p^2_- = v^2_B/v^2_D$,
- $[0.7, 1]$ with step size 2×10^{-2} for p^2_χ,
- $[0.8 \times 10^{-4}, 1.2 \times 10^{-4}]$ with step size 4×10^{-6} for p_{stop}.
- $[0.5, 0.6]$ with step size 2×10^{-2} for r

We find that the optimal mixture is driven by $r = 54\%$ for the first set of parameters: $p^1_\chi = 0.26$, $v^1_B/v^1_D = 0.86$, and $1 - r = 46\%$ for the second one: $p^2_\chi = 0.86$, $v^2_B/v^2_D = 0.58$ and p_{stop}=1.04×10^{-4}.

Fig. 6.9 shows the induced split in the space of parameters. The first thing to notice is that this is a 'real' mixture, in the sense that the two modes are very distinct, and their respective weights are similar. In particular, the recognition probabilities become very different in both modes, and different from the initial model and the *in vitro* estimates (Dixon & Kowalczykowski, 1993b; Taylor & Smith, 1992a, 2003).

Fig. 6.10 shows the marked improvement on fitting for the first peak (at the position of the first Chi). The improvement is noticeable both in the proximal region, where the high-recognition mode allows the model to fit better the initial peak (solid line), and compares well with the initial fit (dotted line); and, at the far end, where the low-recognition mode delineates the finer details of the data better as well (one sees the presence of the two Chi sites clearly in the prediction). This is confirmed by the BIC scores (see Table 2) that indicate a strong preference for the mixture model.

It is instructive to compare the predicted mean values of τ_1 for all four parameter sets (including the one coming from *in vitro* estimates (Taylor & Smith, 2003). If we compare these mean values at $60kb$ we get:

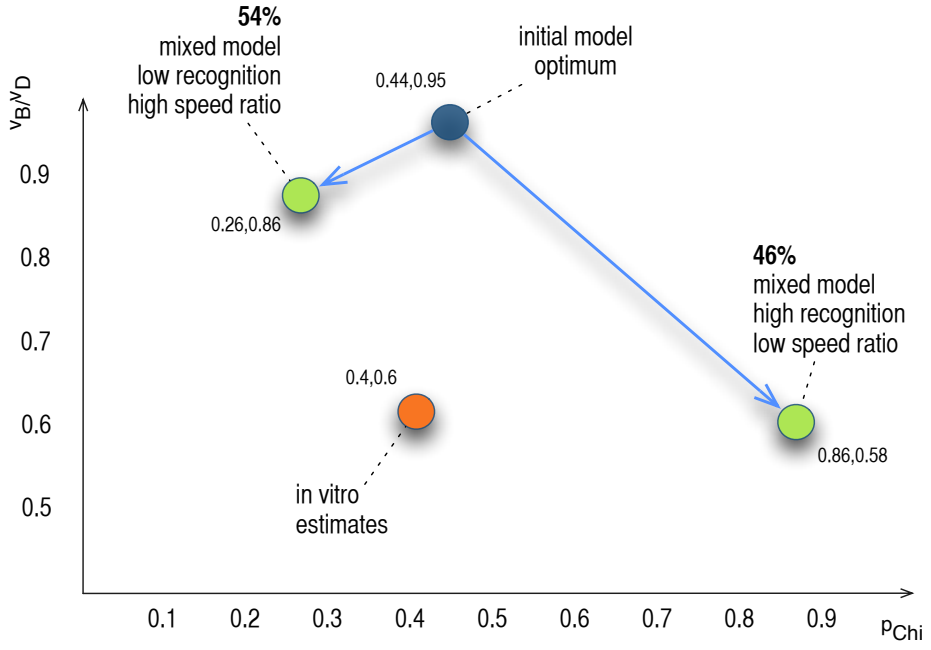


Figure 6.9. The two parameter sets in the mixture model compared to the optimal parameters of the initial model. Percentages indicate the probabilities of the low-recognition/high-ratio mode (46%), and of the high-recognition/low-ratio mode (54%).

in vitro estimates: $60\frac{4}{6} \sim 40kb$

initial model: $60\frac{5}{95} \sim 3kb$

mixture model:

- *low recognition mode* $60\frac{14}{86} \sim 9kb$
- *high recognition mode* $60\frac{42}{58} \sim 43kb$

At first, the prediction for the high recognition mode of the mixed model ($45kb$) seems improbably long. However, it is important to note that this model predicts a very high probability of Chi recognition. Given that Chi sites are present on average every $5kb$ on the chromosome (Touzain *et al.*, 2010), in this mode RecBCD would very rarely

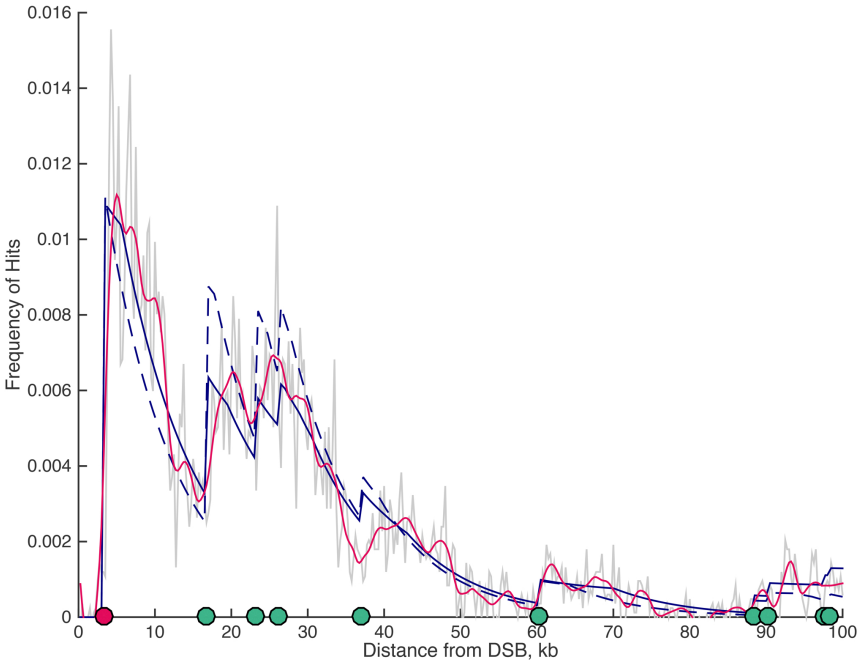


Figure 6.10. Comparison of the 1 Chi data set and the predictions of the optimal mixed model (solid line, $p_{stop} = 1.04 \times 10^{-4}, p_\chi^1 = 0.26, v_B^1/v_D^1 = 0.86, p_\chi^2 = 0.86, v_B^2/v_D^2 = 0.58, r = 54\%$), and the optimal basic one (dotted line, $p_{stop} = 1.12 \times 10^{-4}, p_\chi = 0.44, v_B^1/v_D = 0.95$). The Chi sites are depicted by green circles except for the position of the Chi array which is in red. The grey line shows the raw data binned into 250 bp bins. The red curve represent the smoothed data with a 'loess' filter (bandwith 5700, span 0.057).

travel such a large distance before Chi recognition. One can calculate the mean τ_1 over all Chi sites, assuming a Chi site every $5kb$ and taking into account the probability of Chi recognition: both the high and low recognition modes of the mixed model have a mean τ_1 of the same order of magnitude ($5\frac{14}{86}\frac{100}{26} \sim 3.1kb$ and $5\frac{42}{58}\frac{100}{86} \sim 4.2kb$). This value is significantly higher than that predicted from the initial model ($5\frac{5}{95}\frac{100}{44} \sim 0.6kb$). It might be that a minimal loop size is

important to ensure efficient loading of RecA at Chi which could be reflected in the predictions of the mixed model.

Table 6.2. Comparison of the mixture model and the basic model for the data set with 1 Chi site in the Chi-array. The BIC scores have been computed using Eq. 6.10

BIC basic model(v_B/v_D estimated)	BIC mixture model	Preferred model
92087	91735	mixture model (very strong)

As all molecular systems, the double-strand break repair system is faced with trade-offs. The density of Chi sites found on the chromosome together with the imperfect recognition thereof could be interpreted as a sign that recognition accuracy is traded off against some additional desirable properties. Such properties could be: speed of execution of the resection, optimisation of the length of the segment on which RecA will be loaded and the search for homology will be based (Forget & Kowalczykowski, 2012), control of the variance of this length. Taking these new quantitative insights into account, and insofar as the model captures well the general features of the hit counts, and their dependency on the variations of the Chi distributions, one can use it as a quantitative tool in the investigation of the reasons for the genomic distribution of Chi sites (Touzain *et al.*, 2010). Specifically, one can ask whether this distribution is judiciously adjusted to the generation of a resected single strand which optimises the performance of the RecA-based homology search and hence of the entire DSB repair process. The hypothetic single molecule *in vivo* bimodal behaviour, which our data-driven model suggests, would avail the cell with a larger palette of repair options, and thus should be integral to this investigation.

6.4 Model availability

The model is available at https://github.com/milanafilatenkova/RecBCDmodel.

Chapter 7

Discussion

In this thesis I presented a Markov Chain framework combined with statistical inference to perform quantitative analysis of ChIP-Seq data. The primary purpose of this method is to reverse engineer the stochastic behaviour of a DNA binding molecular machine from its population scale trace obtained from ChIP-Sequencing.

In order to demonstrate how this method works in practice I applied it to *in vivo* characterisation of RecBCD mediated RecA binding in the vicinity of a DSB. The structure of the MC model and its parametric formulation were designed by incorporating previous knowledge about RecBCD activity *in vitro*. With the help of classical statistical inference I tested the model by fitting it to ChIP-Seq data and quantified the key parameters governing RecBCD activity.

This unique combination of genomic and mechanistic modelling can be utilised to infer characteristics of other DNA processing enzymes *in vivo* such as RNA polymerase, for example, provided high resolution sequence data is available.

It is worth noting that the most important ingredient of the ChIP-Seq modelling recipe proposed in this thesis is coarse-grained understanding of the mechanism of the process that generates the data observed. Some preliminary knowledge about the system of interest should be available to build a Markov Chain in the manner explained in this thesis. Fitting a Markov Chain model to the fragment count distribution would only help to confirm or refine the mechanism proposed elsewhere. This type of modelling can be applied to clarify a molecular process that has been previously studied *in vitro*, for example.

As direct visualisation of the DNA binding proteins is still beyond experimental reach, the method developed in this thesis provides an alternative indirect way of *in vivo* characterisation of DNA binding proteins using ChIP-Seq.

7.1 Inference of the parameters of RecBCD action *in vivo*

In Chapter 6 I demonstrated that a simple MC model with only three parameters is enough to attain a good fit to the ChIP-Seq data of RecBCD dependent RecA binding.

Initially I assumed only a single mode of action of RecBCD. The parameter estimates were significantly deviating from those reported *in vitro* ($VB/VD^{in\ vivo} = 0.9 > VB/VD^{in\ vitro} = 0.6$ (Taylor & Smith, 2003)). Since the estimated transition probabilities from *in vivo* data should be seen as those reflective of real physiological conditions inside the cell during the process of repair, the difference might be attributable to conditions in the cell different from those in a test tube, which affected the absolute speeds of the motors and consequently their

ratio. Alternatively, Adar *et al.* (2004), for example, suggested that the transition probabilities of SA should be a function of the conditions of the medium. In our case the probabilities of transitions could also be set as a function of the concentrations of ATP and Mg^{2+}, for example, but that would be the topic of a new research project.

Although the motor speed appears to be different *in vivo*, the probability of Chi recognition and processivity were estimated to be similar to those measured *in vitro* (Dixon & Kowalczykowski, 1993a; Taylor & Smith, 1992b).

7.2 Limitations

In order to construct the model of RecBCD activity on the DNA I made several assumptions. First of all it was assumed that RecA (the readout protein of RecBCD sequence output) covers the entire sequence segment produced by RecBCD (OS). The parameter estimates are sensitive to this assumption, particularly p_s. Strictly speaking we estimate the probability of termination of RecA polymerisation and not the processivity of RecBCD. If RecA polymerisation does not keep up with the speed of RecBCD after Chi recognition, the single strand would be partly uncovered and the processivity of RecBCD underestimated. Therefore, p_{stop} inferred here is to be understood as an "effective processivity of RecA loading by RecBCD" which is the combination of its DNA unwinding and RecA loading activities (Cockram *et al.*, 2015).

7.3 Heterogeneity

As stated in Section 2.1.10 RecBCD manifests heterogeneous behaviour with respect to its speed of translocation along the DNA depending on its conformation adopted at the time of binding as concluded by Liu *et al.* (2013a). This study reported the existence of two broad populations of RecBCD molecules with different velocities. A wide distribution of RecBCD speeds has also been observed *in vitro* (Taylor & Smith, 2003). In this thesis I tested the possibility of two modes of action of RecBCD with different ratio of the slow to fast motor speeds. The data under a two-population hypothesis turned out to be more likely than a single population model confirming the bimodal kinetic behaviour of RecBCD. Also, the model predicted that the two types of molecules in the mixed population responded to Chi site significantly differently, the slow population being more successful in Chi recognition (higher estimated probability of Chi recognition) than the fast one. Remarkably, both kinetic modes were found to contribute equally to the distribution. Interestingly, the estimated parameters suggest that approximately the same length of the loop prior to Chi recognition would be produced by both types of molecules which may indicate that the length of the loop formed ahead of RecBCD is an important attribute of the function of RecBCD (Cockram *et al.*, 2015). It is worth noting that the results only confirm that the two-population model is preferred to the single-population model, yet we cannot exclude the presence of additional factors disregarded by the model. However, because the mixed model fits the data very well, if some additional factors were missed out their contribution would not be significant.

7.4 Background

One of the drawbacks of the model developed in this thesis is a somewhat simplistic method of estimating the background: it is based on the assumption that there is a region where there is no meaningful signal and the mean hit count number over that region is taken as the background level. Because there was no bias clearly seen in the input when analysing ChIP-Seq data of RecA binding (Section 6.3.1) it was decided to model the background as a uniform signal, though that might also be an oversimplification.

7.5 Noise

Differential models of ChIP-Seq data are important in order to quantify absolute enrichment including the characterisation of the noise in ChIP-Seq data. Different ChIP-Seq tools exploit various models to describe the data. If fragments were independently sampled from a fragment pool, then the hit counts would follow a multinomial distribution, which can be approximated by the Poisson distribution provided the total number of counts is sufficiently large (Poisson approximation of Multinomial). Poisson model has been implemented by some ChIP-Seq analysis tools like MACS (Zhang *et al.*, 2008a), PeakSeq (Rozowsky *et al.*, 2009a) and by Diaz *et al.* (2012). Poisson model though is subject to false discoveries induced by overdispersion and zero-inflation. The reason for that is the fact that Poisson distribution does not capture the observed overdispersion in the data (Diaz *et al.*, 2012; Nagalakshmi *et al.*, 2008; Robinson & Smyth, 2007).

Within my framework I assume that the signal follows a multinomial distribution (approximated by Poisson) despite obvious overdispersion in the signal that this

model fails to explain. Since the focus of this study is to characterise the main trend in the data rather than the noise I stick to this conventional multinomial model, since overdispersion should not affect the average count and hence the estimates of the parameters. Contrary to peak detecting algorithms where the protein binding is scarce and the signal to background ratio is very low and therefore failure to discriminate the signal from noise can lead to false discoveries, in our case the binding is continuous and the signal is large compared to the noise, so it is the average relative count rather than the absolute count that is of interest. In the absence of proper account for the source of overdispersion in ChIP-Seq data this is as much as can be done at this stage.

The Negative Binomial (NB) distribution is commonly used to model count data with overdispersion, for example in SAGE (NB) (Robinson & Smyth, 2007), RNA-Seq (NB)(Robinson *et al.*, 2010), BayesPeak (Spyrou *et al.*, 2009), MOSAiCS (Kuan *et al.*, 2009). The Negative Binomial distribution for modelling of ChIP-Seq data allows to better discriminate between the background and the meaningful signal thus reducing the risk of false peak discoveries. It is worth investigating further whether the models such as NB that account for overdispersion in the signal are better at capturing the noise in the signal.

The NB model imposes uncertainty on the expected count so the variance in the signal now exceeds the mean. The NB model depends on two parameters μ - expectation of the observed count and σ^2 - variance of the observed count. The model assumes that for each genomic position i there is a unique true set of μ_i and σ_i^2 which can be estimated from the data, the number of counts for i modelled as $Y_i \propto NB(Mp_i, \phi_i)$ (Robinson *et al.*, 2010), where M is the library size or total number of reads in the sample, relative abundance of gene i, ϕ_i - parameter of dispersion. In this framework, mean count ($\mu_i = Mp_i$) and variance in the NB

model are related by

$$\sigma^2 = \mu_i + \phi_i \mu_i^2 \tag{7.1}$$

ϕ is an additional parameter to be estimated from the data.
Although the NB model allows to accommodate overdispersion it still remains empirical and disconnected from the reality of the experiment. More studies modelling ChIP-Seq experiment and the source of noise are needed to properly justify the NB model.

7.6 Fragment size

In ChIP-Seq only a few 5' end read nucleotides are mapped to the reference genome. The mapping size being significantly smaller than the average fragment size ($50 \ll 300$) would contribute to bias in the estimate of the parameter of interest according to the result obtained in Chapter 5. I also demonstrated numerically that as the mapping size approaches the average fragment size the bias disappears and parameter estimates converge to their true value.

Besides, in Chapter 5 an "edge" effect was demonstrated when the edges of the binding regions are represented to a lesser extent than the middle nucleotides in the signal output distorting the hit distribution (Fig. 5.4). The "edge" effect is proportional to the length of the fragment. Therefore, in order to detect binding events (and most importantly their relative frequency) with higher precision one would need to reduce the average size of the selected fragments as well as its variation.

It is however impossible to allow fragments which are too small because there is a limit to the size of mapped tags: very small tags would likely map to multiple

locations. Overall, the conclusion following from this work is that the fragments in the final fragment library should not exceed the optimal tag size.

7.7 Bias in RecA distribution

While I demonstrated no GC bias in the input (Chapter 6) I have not really investigated the bias in RecA binding, yet it has been shown that RecA binds more willingly to the TGG-repeat sequences (Rajan *et al.*, 2006; Tracy & Kowalczykowski, 1996). In Chapter 5 I showed how to estimate the period of GC rich sub-sequences. The GC rich islands would correspond to a higher signal compared to those poor in GC. The data can be filtered to eliminate the effect of this bias by binning or smoothing the data with a moving average larger or equal to the period of GC rich sub-sequences. The same method could be applied to filter out sequence bias of RecA binding created by TGG-repeats. I suggest that as an idea for a future project.

7.8 Removal of identical reads prior to mapping

In Chapter 5 it has been demonstrated that the probability of sampling identical fragments independently increases with the hit frequency. Identical reads appear more frequently by chance as the total number of reads grows. As PCR creates duplicates unevenly across the chromosome, identical reads are normally removed to eliminate sequencing bias. However it is important to appreciate the fact that when the frequency exceeds $\sim$ 10 nt^{-1} removal of identical reads introduces another bias into signal quantification.

7.9 Running average wins over binning

While the majority of ChIP-Seq processing tools use binning to filter out noise this study has demonstrated that a running average is a better way of smoothing the data. As discovered in Chapter 5 a moving average introduces less bias into the parameter estimates as compared to binning. The size of the smoothing window should be chosen as a trade-off between the period of sequence unevenness and the scale of the data to balance data robustness and resolution.

7.10 Extrapolation of the method to other systems

In this thesis I have explored Stochastic Automata - Markov Chain framework for modelling RecBCD and also the analysis of ChIP-Seq profiles of RecBCD-mediated RecA binding in the vicinity of a DSB. I believe this methodology can be extended to other DNA processing molecular machines when attempting to clarify mechanistic details of their activity *in vivo* from ChIP-Seq data. Among those are, for example, SeqA that can polymerise behind the Replication fork, the polymerisation being enhanced by specific sequences (for a review see Touzain *et al.* (2010)). Another example relates to transcription factors interacting with specific promoter sequences (for a review see Spitz & Furlong (2012)). Any other molecular system that stochastically interacts with special motifs on the DNA and generates a measurable sequence output would be a suitable candidate to be analysed using the mathematical framework developed in this thesis.

References

Adar, R., Benenson, Y., Linshiz, G., Rosner, A., Tishby, N. & Shapiro, E. (2004). Stochastic computing with biomolecular automata. *Proceedings of the National Academy of Sciences of the United States of America*, **101**, 9960–9965.

Aird, D., Ross, M.G., Chen, W.S., Danielsson, M., Fennell, T., Russ, C., Jaffe, D.B., Nusbaum, C. & Gnirke, A. (2011). Analyzing and minimizing pcr amplification bias in illumina sequencing libraries. *Genome Biol*, **12**, R18.

Arenson, T.A., Tsodikov, O.V. & Cox, M.M. (1999). Quantitative analysis of the kinetics of end-dependent disassembly of reca filaments from ssdna. *Journal of molecular biology*, **288**, 391–401.

Arnold, D.A., Handa, N., Kobayashi, I. & Kowalczykowski, S.C. (2000). A novel, 11 nucleotide variant of chi, chi*: one of a class of sequences defining the escherichia coli recombination hotspot chi. *J Mol Biol*, **300**, 469–479.

Bailey, T., Krajewski, P., Ladunga, I., Lefebvre, C., Li, Q., Liu, T., Madrigal, P., Taslim, C. & Zhang, J. (2013). Practical guidelines for the comprehensive analysis of chip-seq data. *PLoS Comput Biol*, **9**, e1003326.

Bao, Y., Vinciotti, V., Wit, E. & AC't Hoen, P. (2013). Accounting for immunoprecipitation efficiencies in the statistical analysis of chip-seq data. *BMC bioinformatics*, **14**, 169.

Bar-Ziv, R., Tlusty, T. & Libchaber, A. (2002). Protein–dna computation by stochastic assembly cascade. *Proceedings of the National Academy of Sciences*, **99**, 11589–11592.

Bell, J.C., Plank, J.L., Dombrowski, C.C. & Kowalczykowski, S.C. (2012). Direct imaging of reca nucleation and growth on single molecules of ssb-coated ssdna. *Nature*, **491**, 274–278.

Benenson, Y., Paz-Elizur, T., Adar, R., Keinan, E., Livneh, Z. & Shapiro, E. (2001). Programmable and autonomous computing machine made of biomolecules. *Nature*, **414**, 430–434.

Benenson, Y., Adar, R., Paz-Elizur, T., Livneh, Z. & Shapiro, E. (2003). Dna molecule provides a computing machine with both data and fuel. *Proceedings of the National Academy of Sciences*, **100**, 2191–2196.

Bennett, C.H. (1982). The thermodynamics of computation—a review. *International Journal of Theoretical Physics*, **21**, 905–940.

Bianco, P.R., Brewer, L.R., Corzett, M., Balhorn, R., Yeh, Y., Kowalczykowski, S.C. & Baskin, R.J. (2001). Processive translocation and dna unwinding by individual recbcd enzyme molecules. *Nature*, **409**, 374–378.

Carrasco, C., Dillingham, M.S. & Moreno-Herrero, F. (2014). Single molecule approaches to monitor the recognition and resection of double-stranded dna breaks during homologous recombination. *DNA Repair (Amst)*, **20**, 119–129.

Chen, Z., Yang, H. & Pavletich, N.P. (2008). Mechanism of homologous recombination from the reca–ssdna/dsdna structures. *Nature*, **453**, 489–494.

Cheng, K.C. & Smith, G.R. (1987). Cutting of Chi-like sequences by the RecBCD enzyme of Escherichia coli. *Journal of Molecular Biology*, **194**, 747–750.

Cheng, K.C. & Smith, G.R. (1989). Distribution of chi-stimulated recombinational exchanges and heteroduplex endpoints in phage lambda. *Genetics*, **123**, 5–17.

Chis, O.T., Banga, J.R. & Balsa-Canto, E. (2011). Structural identifiability of systems biology models: a critical comparison of methods. *PloS one*, **6**, e27755.

Churchill, J.J., Anderson, D.G. & Kowalczykowski, S.C. (1999). The recbc enzyme loads reca protein onto ssdna asymmetrically and independently of chi, resulting in constitutive recombination activation. *Genes Dev*, **13**, 901–911.

Cockram, C.A., Filatenkova, M., Danos, V., El Karoui, M. & Leach, D.R. (2015). Quantitative genomic analysis of reca protein binding during dna double-strand break repair reveals recbcd action in vivo. *Proceedings of the National Academy of Sciences*, 201424269.

Cox, M.M. (2007). Regulation of bacterial reca protein function. *Critical Reviews in Biochemistry and Molecular Biology*, **42**, 41–63.

Cox, M.M. (2013). Proteins pinpoint double strand breaks. *eLife*, **2**, e01561.

Diaz, A., Park, K., Lim, D.A. & Song, J.S. (2012). Normalization, bias correction, and peak calling for chip-seq. *Stat Appl Genet Mol Biol*, **11**, Article 9.

Dillingham, M.S. & Kowalczykowski, S.C. (2008). RecBCD enzyme and the repair of double-stranded DNA breaks. *Microbiology and Molecular Biology Reviews*, **72**, 642–671.

Dillingham, M.S., Webb, M.R. & Kowalczykowski, S.C. (2005). Bipolar dna translocation contributes to highly processive dna unwinding by recbcd enzyme. *J Biol Chem*, **280**, 37069–37077.

Dixon, D.A. & Kowalczykowski, S.C. (1993a). The recombination hotspot χ is a regulatory sequence that acts by attenuating the nuclease activity of the e. coli recbcd enzyme. *Cell*, **73**, 87–96.

Dixon, D.A. & Kowalczykowski, S.C. (1993b). The recombination hotspot χ is a regulatory sequence that acts by attenuating the nuclease activity of the e. coli recbcd enzyme. *Cell*, **73**, 87–96.

Drees, J.C., Lusetti, S.L., Chitteni-Pattu, S., Inman, R.B. & Cox, M.M. (2004). A reca filament capping mechanism for recx protein. *Molecular cell*, **15**, 789–798.

Ennis, D.G., Amundsen, S.K. & Smith, G.R. (1987). Genetic functions promoting homologous recombination in escherichia coli: a study of inversions in phage lambda. *Genetics*, **115**, 11–24.

Eykelenboom, J.K., Blackwood, J.K., Okely, E. & Leach, D.R. (2008). Sbccd causes a double-strand break at a dna palindrome in the escherichia coli chromosome. *Molecular cell*, **29**, 644–651.

Forget, A.L. & Kowalczykowski, S.C. (2012). Single-molecule imaging of dna pairing by reca reveals a three-dimensional homology search. *Nature*, **482**, 423–427.

Furey, T.S. (2012). Chip–seq and beyond: new and improved methodologies to detect and characterize protein–dna interactions. *Nature Reviews Genetics*, **13**, 840–852.

Galletto, R., Amitani, I., Baskin, R.J. & Kowalczykowski, S.C. (2006). Direct observation of individual reca filaments assembling on single dna molecules. *Nature*, **443**, 875–878.

Goren, A., Ozsolak, F., Shoresh, N., Ku, M., Adli, M., Hart, C., Gymrek, M., Zuk, O., Regev, A., Milos, P.M. & Bernstein, B.E. (2010). Chromatin profiling by directly sequencing small quantities of immunoprecipitated dna. *Nat Methods*, **7**, 47–49.

Handa, N., Ohashi, S., Kusano, K. & Kobayashi, I. (1997). Chi-star, a chi-related 11-mer sequence partially active in an e. coli recc1004 strain. *Genes Cells*, **2**, 525–536.

Handa, N., Bianco, P.R., Baskin, R.J. & Kowalczykowski, S.C. (2005). Direct visualization of recbcd movement reveals cotranslocation of the recd motor after chi recognition. *Mol Cell*, **17**, 745–750.

Hastings, W.K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, **57**, 97–109.

Hercus, C. (2012). Novoalign. *Selangor: Novocraft Technologies*.

Joo, C., McKinney, S.A., Nakamura, M., Rasnik, I., Myong, S. & Ha, T. (2006). Real-time observation of reca filament dynamics with single monomer resolution. *Cell*, **126**, 515–527.

Kass, R.E. & Raftery, A.E. (1995). Bayes factors. *Journal of the american statistical association*, **90**, 773–795.

Kozarewa, I., Ning, Z., Quail, M.A., Sanders, M.J., Berriman, M. & Turner, D.J. (2009). Amplification-free illumina sequencing-library preparation facilitates improved mapping and assembly of (g+c)-biased genomes. *Nat Methods*, **6**, 291–295.

Kuan, P., Chung, D., Pan, G., Thomson, J., R, S. & S, K. (2009). A statistical framework for the analysis of chip-seq data. *Technical Report.*.

Landt, S.G., Marinov, G.K., Kundaje, A., Kheradpour, P., Pauli, F., Batzoglou, S., Bernstein, B.E., Bickel, P., Brown, J.B., Cayting, P. *et al.* (2012). Chip-seq guidelines and practices of the encode and modencode consortia. *Genome research*, **22**, 1813–1831.

Lesterlin, C., Ball, G., Schermelleh, L. & Sherratt, D.J. (2014). Reca bundles mediate homology pairing between distant sisters during dna break repair. *Nature*, **506**, 249–253.

Li, J., Jiang, H. & Wong, W.H. (2010). Modeling non-uniformity in short-read rates in rna-seq data. *Genome Biol*, **11**, R50.

Liu, B., Baskin, R.J. & Kowalczykowski, S.C. (2013a). Dna unwinding heterogeneity by recbcd results from static molecules able to equilibrate. *Nature*, **500**, 482–485.

Liu, B., Baskin, R.J. & Kowalczykowski, S.C. (2013b). DNA unwinding heterogeneity by recbcd results from static molecules able to equilibrate. *Nature*, **500**, 482–485.

Lovett, S.T. (2012). Biochemistry: A glimpse of molecular competition. *Nature*, **491**, 198–200.

Lucius, A.L., Vindigni, A., Gregorian, R., Ali, J.A., Taylor, A.F., Smith, G.R. & Lohman, T.M. (2002). Dna unwinding step-size of e. coli recbcd helicase determined from single turnover chemical quenched-flow kinetic studies. *Journal of molecular biology*, **324**, 409–428.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. & Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, **21**, 1087–1092.

Myers, R.S., Stahl, M.M. & Stahl, F.W. (1995). Chi recombination activity in phage lambda decays as a function of genetic distance. *Genetics*, **141**, 805–812.

Nagalakshmi, U., Wang, Z., Waern, K., Shou, C., Raha, D., Gerstein, M. & Snyder, M. (2008). The transcriptional landscape of the yeast genome defined by rna sequencing. *Science*, **320**, 1344–1349.

New England Biolabs, Inc. (Version 6.0). *Instruction Manual*.

Rabin, M.O. (1963). Probabilistic automata. *Information and control*, **6**, 230–245.

Rajan, R., Wisler, J.W. & Bell, C.E. (2006). Probing the dna sequence specificity of escherichia coli reca protein. *Nucleic acids research*, **34**, 2463–2471.

Robinson, M.D. & Smyth, G.K. (2007). Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*, **23**, 2881–2887.

Robinson, M.D., McCarthy, D.J. & Smyth, G.K. (2010). edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**, 139–140.

Rohs, R., Jin, X., West, S.M., Joshi, R., Honig, B. & Mann, R.S. (2010). Origins of specificity in protein-dna recognition. *Annual review of biochemistry*, **79**, 233.

Roman, L., Eggleston, A. & Kowalczykowski, S. (1992). Processivity of the dna helicase activity of escherichia coli recbcd enzyme. *Journal of Biological Chemistry*, **267**, 4207–4214.

Rosenthal, J.S. *et al.* (2011). Optimal proposal distributions and adaptive mcmc. *Handbook of Markov Chain Monte Carlo*, 93–112.

Rozowsky, J., Euskirchen, G., Auerbach, R.K., Zhang, Z.D., Gibson, T., Bjornson, R., Carriero, N., Snyder, M. & Gerstein, M.B. (2009a). Peakseq enables systematic scoring of chip-seq experiments relative to controls. *Nat Biotechnol*, **27**, 66–75.

Rozowsky, J., Euskirchen, G., Auerbach, R.K., Zhang, Z.D., Gibson, T., Bjornson, R., Carriero, N., Snyder, M. & Gerstein, M.B. (2009b). Peakseq enables systematic scoring of chip-seq experiments relative to controls. *Nature biotechnology*, **27**, 66–75.

Schwarz, G. *et al.* (1978). Estimating the dimension of a model. *The annals of statistics*, **6**, 461–464.

Shendure, J. & Ji, H. (2008). Next-generation dna sequencing. *Nature biotechnology*, **26**, 1135–1145.

Shivashankar, G., Feingold, M., Krichevsky, O. & Libchaber, A. (1999). Reca polymerization on double-stranded dna by using single-molecule manipulation: the role of atp hydrolysis. *Proceedings of the National Academy of Sciences*, **96**, 7916–7921.

Singleton, M.R., Dillingham, M.S., Gaudier, M., Kowalczykowski, S.C. & Wigley, D.B. (2004). Crystal structure of recbcd enzyme reveals a machine for processing dna breaks. *Nature*, **432**, 187–193.

Smith, G.R. (2012). How recbcd enzyme and chi promote dna break repair and recombination: a molecular biologist's view. *Microbiology and Molecular Biology Reviews*, **76**, 217–228.

Spies, M. & Kowalczykowski, S.C. (2006). The reca binding locus of recbcd is a general domain for recruitment of dna strand exchange proteins. *Mol Cell*, **21**, 573–580.

Spies, M., Bianco, P.R., Dillingham, M.S., Handa, N., Baskin, R.J. & Kowalczykowski, S.C. (2003). A molecular throttle: the recombination hotspot chi controls dna translocation by the recbcd helicase. *Cell*, **114**, 647–654.

Spies, M., Dillingham, M.S. & Kowalczykowski, S.C. (2005). Translocation by the recb motor is an absolute requirement for χ-recognition and reca protein loading by recbcd enzyme. *Journal of Biological Chemistry*, **280**, 37078–37087.

Spies, M., Amitani, I., Baskin, R.J. & Kowalczykowski, S.C. (2007). Recbcd enzyme switches lead motor subunits in response to χ recognition. *Cell*, **131**, 694–705.

Spitz, F. & Furlong, E.E. (2012). Transcription factors: from enhancer binding to developmental control. *Nature Reviews Genetics*, **13**, 613–626.

Spyrou, C., Stark, R., Lynch, A.G. & Tavaré, S. (2009). Bayespeak: Bayesian analysis of chip-seq data. *BMC Bioinformatics*, **10**, 299.

Stukalin, E.B., Phillips III, H. & Kolomeisky, A.B. (2005a). Coupling of two motor proteins: a new motor can move faster. *Physical review letters*, **94**, 238101.

Stukalin, E.B., Phillips III, H. & Kolomeisky, A.B. (2005b). Coupling of two motor proteins: a new motor can move faster. *Physical review letters*, **94**, 238101.

Symington, L.S. & Gautier, J. (2011). Double-strand break end resection and repair pathway choice. *Annual review of genetics*, **45**, 247–271.

Taylor, A.F. & Smith, G.R. (1992a). Recbcd enzyme is altered upon cutting dna at a chi recombination hotspot. *Proceedings of the National Academy of Sciences*, **89**, 5226–5230.

Taylor, A.F. & Smith, G.R. (1992b). Recbcd enzyme is altered upon cutting dna at a chi recombination hotspot. *Proceedings of the National Academy of Sciences*, **89**, 5226–5230.

Taylor, A.F. & Smith, G.R. (1995). Strand specificity of nicking of dna at chi sites by recbcd enzyme modulation by atp and magnesium levels. *Journal of Biological Chemistry*, **270**, 24459–24467.

Taylor, A.F. & Smith, G.R. (2003). Recbcd enzyme is a DNA helicase with fast and slow motors of opposite polarity. *Nature*, **423**, 889–893.

Touzain, F., Petit, M.A., Schbath, S. & El Karoui, M. (2010). DNA motifs that sculpt the bacterial chromosome. *Nature Reviews Microbiology*, **9**, 15–26.

Tracy, R.B. & Kowalczykowski, S.C. (1996). In vitro selection of preferred dna pairing sequences by the escherichia coli reca protein. *Genes & development*, **10**, 1890–1903.

Umezu, K. & Kolodner, R.D. (1994). Protein interactions in genetic recombination in escherichia coli. interactions involving reco and recr overcome the inhibition of reca by single-stranded dna-binding protein. *Journal of Biological Chemistry*, **269**, 30005–30013.

Van Der Heijden, T., Van Noort, J., Van Leest, H., Kanaar, R., Wyman, C., Dekker, N. & Dekker, C. (2005). Torque-limited reca polymerization on dsdna. *Nucleic acids research*, **33**, 2099–2105.

Vilenchik, M.M. & Knudson, A.G. (2003). Endogenous dna double-strand breaks: production, fidelity of repair, and induction of cancer. *Proceedings of the National Academy of Sciences*, **100**, 12871–12876.

Wigley, D.B. (2012). Bacterial dna repair: recent insights into the mechanism of recbcd, addab and adnab. *Nature Reviews Microbiology*, **11**, 9–13.

Wyman, C., Ristic, D. & Kanaar, R. (2004). Homologous recombination-mediated double-strand break repair. *DNA repair*, **3**, 827–833.

Yang, L., Handa, N., Liu, B., Dillingham, M.S., Wigley, D.B. & Kowalczykowski, S.C. (2012). Alteration of recognition by recbcd reveals a regulated molecular latch and suggests a channel-bypass mechanism for biological control. *Proc Natl Acad Sci U S A*, **109**, 8907–8912.

Zhang, Y., Liu, T., Meyer, C.A., Eeckhoute, J., Johnson, D.S., Bernstein, B.E., Nusbaum, C., Myers, R.M., Brown, M., Li, W. & Liu, X.S. (2008a). Model-based analysis of chip-seq (macs). *Genome Biol*, **9**, R137.

Zhang, Z.D., Rozowsky, J., Snyder, M., Chang, J. & Gerstein, M. (2008b). Modeling chip sequencing in silico with applications. *PLoS Comput Biol*, **4**, e1000158.

Appendix A

Published papers

Cockram, C.A., Filatenkova, M., Danos, V., El Karoui, M. & Leach, D.R. (2015). Quantitative genomic analysis of reca protein binding during dna double-strand break repair reveals recbcd action in vivo. Proceedings of the National Academy of Sciences, 201424269.